of Engineers
Construction Engineering
Research Laboratory

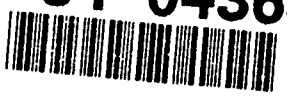
USACERL Special Report N-91/22
May 1991

Engineering Applications of Neural Computing: A State-of-the-Art Survey

by
Xiping Wu
James D. Westervelt

Neural computing, as a paradigm of knowledge representation and information processing, has attracted tremendous enthusiasm and research interest recently. With advancing sophistication, neural computing technology has been successfully tailored for a wide range of applications, including some engineering fields. With the development of hardware-based neural networks and neural computing theory, neural networks potentially provide efficient tools for solving some difficult engineering problems related to U.S. Army Construction Engineering Research Laboratory (USACERL) research.

This report reviews and describes different types of neural networks including feedforward, feedback, and recurrent networks, their learning algorithms and recent developments. The emphasis is on the most frequently used multilayer feedforward neural networks. Representative publications on neural network applications to engineering problems related and/or of interest to research at USACERL, especially civil engineering problems, are also covered, and each modeling methodology identified. An extensive reference list is provided with each subject. The appendices list major technical journals dedicated to the theory and application of neural computing, some publicly available neural network simulators, and selected books and proceedings published on neural networks and genetic algorithms.



REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave Blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	May 1991	Final

4. TITLE AND SUBTITLE

Engineering Applications of Neural Computing: A State-of-the-Art Survey

5. FUNDING NUMBERS

PE 4A161102
PR AT23
TA NN
WU EF0

6. AUTHOR(S)

Xiping Wu and James D. Westervelt

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

USACERL
PO Box 9005
Champaign, IL 51826-9005

8. PERFORMING ORGANIZATION REPORT NUMBER

USACERL SR N-91/22

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

HQUSACE
20 Massachusetts Avenue, NW.
Washington, DC 23140-1000

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

Copies are available from the National Technical Information Service, 5285 Port Royal Road, Springfield, VA 22161.

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

Neural computing, as a paradigm of knowledge representation and information processing, has attracted tremendous enthusiasm and research interest recently. With advancing sophistication, neural computing technology has been successfully tailored for a wide range of applications, including some engineering fields. With the development of hardware-based neural networks and neural computing theory, neural networks potentially provide efficient tools for solving some difficult engineering problems related to U.S. Army Construction Engineering Research Laboratory (USACERL) research.

This report reviews and describes different types of neural networks including feedforward, feedback, and recurrent networks, their learning algorithms and recent developments. The emphasis is on the most frequently used multilayer feedforward neural networks. Representative publications on neural network applications to engineering problems related and/or of interest to research at USACERL, especially civil engineering problems, are also covered, and each modeling methodology identified. An extensive reference list is provided with each subject. The appendices list major technical journals dedicated to the theory and application of neural computing, some publicly available neural network simulators, and selected books and proceedings published on neural networks and genetic algorithms.

14. SUBJECT TERMS

neural computing
state of the art

15. NUMBER OF PAGES

140

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	SAR

FOREWORD

This report was funded by Headquarters, U.S. Army Corps of Engineers, under project 4A161102AT23, "Basic Research in Military Construction"; Task Order NN, "Neural Network Analysis"; Work Unit EF0, "Cognitive Processes."

Xiping Wu is a graduate research assistant from the Department of Civil Engineering at the University of Illinois, Urbana-Champaign. Appreciation is given to Professor J. Ghaboussi, Department of Civil Engineering, University of Illinois and to Professor J.H. Garrett, Jr., Department of Civil Engineering, Carnegie-Mellon University for the help and guidance provided during this study. The work was monitored by the Environmental Division (EN) of the U.S. Army Construction Engineering Research Laboratory (USACERL). James Westervelt, USACERL-EN, was the principal investigator. Dr. Edward W. Novak is the acting Chief, USACERL-EN. Technical editing was provided by Linda L. Wheatley, USACERL Information Management Office.

COL Everett R. Thomas is the Commander and Director of USACERL, and Dr. L.R. Shaffer is the Technical Director.

CONTENTS

FIGURES

ENGINEERING APPLICATION OF NEURAL COMPUTING: A STATE-OF-THE-ART SURVEY

1 INTRODUCTION

1.1 Background

In the past 5 years, neural networks, growing out of McCulloch and Pitts' computational model of neuron (McCulloch and Pitts, 1943) and propelled by the works of Hopfield (Hopfield, 1982) and Parallel Distributed Processing (PDP) research group (Rumelhart and McClelland, 1986), have attracted tremendous enthusiasm and research interest from computer scientists, neurophysiologists, and engineers. This remarkable phenomenon has paralleled research in Artificial Intelligence (AI) of the 1970s. This new interest is supported by the realization that neural computing is inherently parallel and functionally more close to the operation of the brain; that is, it has the capability of self-organization or learning. In addition to the recognition of the capability of neural networks and the development of computing technology, other factors have contributed to the explosion of interest in this area: 1) it is a universal approximator or it is computationally complete, which means that an appropriate neural network with appropriate training rules has the capability of solving virtually any computational tasks; 2) it takes a middle ground between traditional mathematical approach and symbolic AI approach by using numerical methods for learning and expansive representation schemes, as well as adopting a functional use of experimental knowledge; 3) it provides an alternative with efficient performance in solving problems that are currently difficult for a conventional approach, such as speech and natural language process, vision and image analysis, and pattern recognition with the recent insight into algorithms that improve the learning ability of a neural network; 4) it may provide some insight into the computational characteristics of the brain; and 5) it is also intrinsically feasible for the implementation of neural networks to massively parallel hardwares (Aleksander, 1989; Barto, 1989).

Research interest and the increasing funding in neural networks have generated numerous kinds of architecture and learning paradigms. With the advance and sophistication in some branches of neural computing, the technology has been successfully tailored for a wide range of applications, such as the modeling of cognitive process, language understanding, and pattern recognition, as illustrated by the large range of subjects covered in papers appearing in conferences on neural networks (IEEE, 1987, 1988; NIPS, 1988, 1989; IJCNN, 1989, 1990). To facilitate the development and application of this emerging technology, National Science Foundation (NSF) also established the Neural Engineering Program in conjunction with other funding agencies such as Defense Advance Research Projects Agency (DARPA), Office of Naval Research (ONR), and National Aeronautics and Space Administration (NASA). It is obvious that with the advance and development of hardware and software of neural networks, this new technology will potentially solve some difficult engineering problems. Research in application of neural computing to engineering problems, especially civil engineering problems related to U.S. Army Construction Engineering Research Laboratory

(USACERL) research, is still in its fledgling stage. Therefore, it is imperative to discern and evaluate the state-of-the-art of neural computing technology and its practical applications in order to provide insight into the potential opportunities and benefits in USACERL-related research.

1.2 Objectives and Approach

The main objective of this report is to provide a series of short descriptions of how neural networks have been used in fields related to USACERL research and to provide extensive references on seminal works in neural computing. The approach was as follows:

- Review new development and current research on different types of neural networks and learning paradigms.
- Review publications on neural networks applications to engineering problems related to and/or of interest to USACERL research projects. Identify the methodologies of neural computing to different applications and evaluate their potential for research projects in USACERL.
- Provide a comprehensive set of bibliographic references on seminal and representative publications in neural computing.

References

1. *Proceedings of the IEEE First International Conference on Neural Networks*, Institute of Electrical and Electronic Engineers (IEEE), New York, June 1987.
2. *Proceedings of the IEEE International Conference on Neural Networks*, Institute of Electrical and Electronic Engineers (IEEE), New York, June 1988.
3. *Proceedings of the 1988 Connectionist Models Summer School*, D. Touretzsky, G. Hinton, and T. Sejnowski (Eds.), Carnegie Mellon University, Morgan Kaufmann Publishers, San Mateo, CA, 1989.
4. *Proceedings of the 1988 IEEE Conference on Neural Information Processing Systems – Natural and Synthetic: Advances in Neural Information Processing Systems 1*, D. S. Touretzky (Ed.), Morgan Kaufmann Publishers, San Mateo, CA, 1989.
5. *Proceedings of the 1989 IEEE Conference on Neural Information Processing Systems – Natural and Synthetic: Advances in Neural Information Processing Systems 2*, D. S. Touretzky (Ed.), Morgan Kaufmann Publishers, San Mateo, CA, 1990.
6. *Proceedings of the International Joint Conference on Neural Networks*, Co-sponsored by IEEE and the International Neural Network Society (INNS), Washington, D. C., 1989.
7. *Proceedings of the International Joint Conference on Neural Networks*, Co-sponsored by IEEE and the International Neural Network Society (INNS), Washington, D. C., 1990.

8. *Proceedings of the International Joint Conference on Neural Networks*, Co-sponsored by IEEE and the International Neural Network Society, San Diego, 1990.

9. *Proceedings of the Third International Conference on Genetic Algorithms*, J. D. Schaffer (Ed.), Morgan Kaufmann Publishers, San Mateo, CA, 1990.

10. *AIP Conference Proceedings 151: Neural Networks for Computing*, J. S. Denker (Ed.), American Institute of Physics (AIP), Snowbird, UT, 1986.

11. Aleksander, I. (Ed.), *Neural Computing Architecture*, The MIT Press, Cambridge, MA, 1989.

12. Barto, A. G., *Connectionist Learning for Control: An Overview*, COINS Technical Report 89-89, Dept. of Computer and Information Science, University of Massachusetts, Amherst, MA, 1989.

13. Hopfield, J. J., "Neural Networks and Physical Systems with Emergent Collective Computational Abilities," *Proceedings of the National Academy of Sciences*, 79, 2554-2558, 1982.

14. McCulloch, W. S., and Pitts, W., "A Logical Calculus of the Ideas Immanent in Nervous Activity," *Bulletin of Mathematical Biophysics*, 5, pp. 115-133, 1943.

15. Rumelhart, D., and McClelland, J. (Eds.), *Parallel Distributed Processing: Explorations in the Microstructure of Cognition; Vol. 1: Foundations*, The MIT Press, Cambridge, MA, 1986.

2 NEURAL NETWORKS

2.1 Introduction

Neural networks, also referred as Parallel Distributed Processing systems (PDP) and Connectionist systems, are computational methods inspired or loosely modelled after the structure and internal information processing operations of the human brain. In general, according to Rumelhart, Hinton and McClelland (1986), a neural network is made up with the following components:

- Processing Units (neurons): the functions of each neuron are to receive signals, perform simple computation, and send out signals through ongoing connections.
- Connections: each connection between neurons functions like a multiplicative filter with its connection strength or weight.
- Rules of Activation Propagation.
- Rules of Learning.
- Network Architecture (Topology).

Based on the composition of the network topology, the form of activation functions and the rule of learning, different kinds of neural networks have been developed. Some of the well-known networks are Perceptrons (Rosenblatt, 1962), Adaline and Madaline (Windrow and Hoff, 1960), Hopfield network (Hopfield, 1982), Boltzman Machine (Hinton, et al., 1983, 1984), Kohonen self-organizing network (Kohonen, 1984), the competitive learning network (Grossberg, 1976; Rumelhart and Zipser, 1986), the Adaptive Resonance Theory (ART) (Carpenter and Grossberg, 1987), the recurrent type networks (Jordan, 1986; Elman, 1988; Williams and Zipser, 1989), and the Backpropagation networks (Rumelhart, et al., 1986; Parker, 1982; Warbos, 1974). In general, learning mechanisms can be categorized into three forms: supervised learning, unsupervised learning, and reinforced learning (Hinton, 1989).

One of the important features pertaining to neural networks is their capability of self-organization or learning. When training a neural network, it is presented with examples or data of the concept to capture. It then internally modifies its interconnection strength or weight of connections through the rule of learning. After completion of the training session, the knowledge is stored in the pattern of connection strengths of the network.

In the following paragraphs, different types of neural networks and their characteristics are described. Relevant references are also included. It should be kept in mind that the references listed here are by no means exhaustive. Our intention is to provide pointers for those interested in neurocomputing application to fundamental works and noticeable achievements as well as the state-of-the-art research in different branches of connectionist systems.

General References

1. Ackley, D. H., Hinton, G. E., and Sejnowski, T. J., "A Learning Algorithm for Boltzmann Machines," *Cognitive Science*, 9, pp. 147-169, 1985.

2. Amari, S.-I., "Neural Theory of Association and Concept Formation," *Biological Cybernetics*, 26, pp. 175-185, 1977.

3. Anderson, J. A., "A Simple Neural Network Generating An Interactive Memory," *Mathematical Biosciences*, 14, pp. 197-220, 1972.

4. Arbib, M. A., *Brains, Machines and Mathematics*, 2nd edition, Springer-Verlag, New York, 1987.

5. Carpenter, G. A., and Grossberg, S., "A Massively Parallel Architecture for a Self-Organizing Neural Pattern Recognition Machine," *Computer Vision, Graphics, and Image Processing*, 37, pp. 54–115, 1987.

6. Carpenter, G. A., and Grossberg, S., "ART 2: Self-Organizing of Stable Category Recognition Codes for Analog Input Patterns," *Applied Optics*, 26, pp. 4919–4930, 1987.

7. Elman, J. L., *Finding Structure in Time*, Technical Report (CRL Tech. Report 8801), University of California, San Diego, Center for Research in Language, La Jolla, CA, 1988.

8. Grossberg, S., "Adaptive Pattern Classification and Universal Recoding: I. Parallel Development and Coding of Neural Feature Detectors," *Biological Cybernetics*, 23, pp. 121-134, 1976.

9. Hebb, D. O., *The Organization of Behavior*, New York, Wiley, 1949.

10. Hinton, G. E., and Anderson, J. A. (Eds.), *Parallel Models of Associative Memory*, Erlbaum, Hillsdale, NJ, 1981.

11. Hinton, G. E., "Connectionist Learning Procedures," *Artificial Intelligence*, 40, pp. 185–234, 1989.

12. Hopfield, J. J., "Neural Networks and Physical Systems with Emergent Collective Computational Abilities," *Proceeding of the National Academy of Sciences*, 79, pp. 2554-2558, 1982.

13. Jordan, M. I., *Serial Order: A Parallel Distributed Processing Approach*, Technical Report (ICS Report 8604), University of California, San Diego, Institute of Cognitive Science, 1986.

14. Kirkpatrick, S., Gelatt, C. D., and Vecchi, M. P., "Optimization by Simulated Annealing," *Science*, 220, pp. 671-680, 1983.

15. Kohonen, T., "Correlation Matrix Memories," *IEEE Transactions on Computers*, C-21, pp.353-359, 1972.

16. Kohonen, T., *Self-organization and Associative Memory*, Springer-Verlag, New York, Second Edition, 1988.

17. Fukushima, K., Miyake, S., and Ito, T., "Neocognitron: a Neural Network Model for a Mechanism of Visual Pattern Recognition," *IEEE Transactions on Systems, Man, and Cybernetics SMC–13*, pp. 826–834, 1983.

18. McCulloch, W. S., and Pitts, W., "A Logical Calculus of the Ideas Immanent in Nervous Activity," *Bulletin of Mathematical Biophysics*, 5, pp. 115-133, 1943.

19. Minsky, M., and Papert, S., *Perceptrons*, The MIT Press, Cambridge, MA, 1969; also the expanded Edition, 1988.

20. Rosenblatt, F., "The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain," *Psychological Review*, 65, pp. 368-408, 1958.

21. Rosenblatt, F., *Principles of Neurodynamics*, Spartan Books, New York, 1962.

22. Rumelhart, D., and Zipser, D., "Feature Discovery by Competitive Learning," *Cognitive Scence*, 9, pp. 75-112, 1985.

23. Rumelhart, D., and McClelland, J. (Eds.), *Parallel Distributed Processing: Explorations in the Microstructure of Cognition; Vol. 1: Foundations*, The MIT Press, Cambridge, MA, 1986.

24. Sejnowski, T. J., and Rosenberg, C. H., "NETalk: A Parallel Network That Learns to Read Aloud," *Report JHU/EECS - 86/01*, The Johns Hopkins University, 1986.

25. Von Newmann, J., *The Computer and the Brain*, New Haven, Yale University Press, 1958.

26. Widrow, B., and Hoff, M. E., "Adaptive Switching Circuits," *1960 IRE WESCON Convention Record*, New York, pp. 96-104, 1960.

27. Williams, R. J., and Zipser, D., "Real-time Recurrent Learning Algorithm," *Connection Science*, 1(1), 1989.

2.2 Feedforward Multilayer Neural Networks

Feedforward neural networks, developed from Perceptron (Rosenblatt, 1958), are also referred to as multilayer perceptrons. It may be claimed that the revival of neural networks research is closely related to the development of backpropagation neural networks and the famous Generalized Delta Rule (Rumelhart, et al., 1986). Because of their simplicity in learning and architecture construction, as well as a sense of control over the training process, backpropagation neural networks have been widely used in most of the applications involved with functional representation and mapping. However, one of the drawbacks associated with multilayer feedforward networks is their slow convergence rate in learning and the lack of a priori determination of architecture and the use of a priori knowledge.

Recently, several approaches have been proposed to improve the performance of backpropagation neural networks. In general, there are these approaches to the problem: 1) using a better representation scheme for input and output, 2) employing higher order learning algorithms other than the gradient descent method, such as the quasi-Newton methods, 3) applying numerical techniques to preprocess the input pattern to introduce independency into the input space, 4) designing innovative training schemes so that certain knowledge is preoriented in the network before final training session, 5) incorporating network geometry adaptation with efficient learning algorithm to build a robust modeling environment, and 6) determining the architecture and training with heuristic rules. The following paragraphs describe perceptron, backpropagation networks, and their variants.

2.2.1 Perceptron

The elementary perceptron is a two-layer feedforward neural networks consisting of the input layer and output layer with the input units being fully connected to output units (Rosenblatt, 1958). The network is basically a heteroassociative, nearest-neighbor pattern matcher in that it maps input patterns presented at the input layer straightforwardly to output patterns at output layer. Connection weights in a perceptron are adjusted using the Perceptron Convergence Theorem (Rosenblatt, 1962) or the Delta Rule, and the activation function is the standard step function. Rosenblatt (1962) has shown that the perceptron can solve a large number of linear mapping problems. However, due to the lack of a hidden layer for intermediate relation representation, the perceptron failed to handle nonlinear separable problems such as the encoding of the exclusive-or (XOR) function, and has poor generalization capability. The critical analysis of perceptron by Minsky and Papert (1969) with regard to its limited mapping capability and the lack of powerful training algorithm for multilayer perceptrons at that time in some way halted the development in neural network research in the 70's.

The training of a perceptron is an iterative process and the algorithm is:

- Randomly initialize the weights and threshold value at each units;
- Compute the output at each output unit:

  $$o_j = f\ (\sum_i w_{ij}\ a_i - \theta_j\ )$$

  where f is a step activation function, θ_j the threshold value, a_i the input activation, o_j the output activation, and w_{ij} represents connection strength between input node i and output node j.
- Adjust weight using the Delta Rule: $\Delta w_{ij} = \eta\ (t_j - o_j)\ a_i$, where η is the learning rate and t_j is the expected output and o_j is the network prediction.

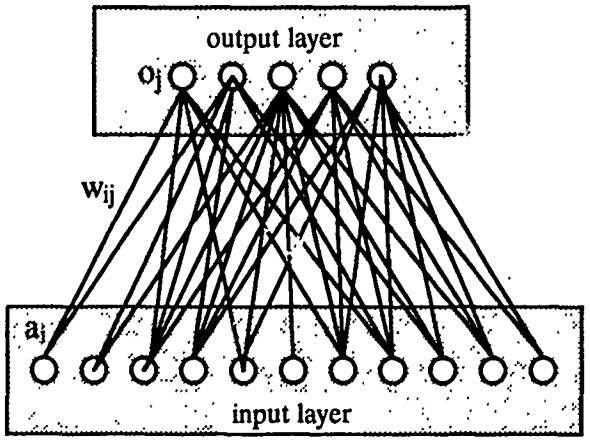


Figure 2.1 – A Sample Perceptron

Because of the limited learning capability, the perceptron is mostly suited for linear mapping problems with binary outputs. Recent research has shown that the capability of the perceptron can be improved by incorporating the mechanism of higher order connectivity or higher order correlations, and fuzzy logic (Maxwell, Giles and Lee, 1986; Keller and Hunt, 1985).

References

1. Keller, J., and Hunt, D., "Incorporating Fuzzy Membership Functions into the Perceptron Algorithm," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-7, pp. 693-699, 1985.

2. Maxwell, T., Giles, C., and Lee, Y., "Nonlinear Dynamics of Artificial Neural Systems," in *AIP Conference Proceedings 151: Neural Networks for Computing*, J. Denker (Ed.), New York, American Institute of Physics, 1986.

3. Minsky, M., and Papert, S., *Perceptrons*, The MIT Press, Cambridge, MA, 1969; also the expanded Edition, 1988.

4. Rosenblatt, F., "The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain," *Psychological Review*, 65, pp. 368-408, 1958.

5. Rosenblatt, F., *Principles of Neurodynamics*, Spartan Books, New York, 1962.

2.2.2 Backpropagation Neural Networks

Backpropagation neural networks are multilayer feedforward networks with the Generalized Delta Rule as their learning rule (Parker, 1982; Rumelhart, et al., 1986; Warbos, 1974). Learning in a backpropagation network is supervised learning which means that the expected output is included in the training data that the network is supposed to learn. The architecture for all backpropagation networks is in a layered form consisting of input layer, output layer, and one or more hidden layers.

The training process via the generalized delta rule is an iterative process. Each training cycle includes two sessions: forward propagation of signals from input to output layer; and backward propagation of error signals seen at the output layer. Thus each cycle involves the determination of error associated with each output units and the modification of weights on the network connections. The learning capacity of a backpropagation network depends on the number of nodes in the hidden layer(s) and the pattern of connections between nodes in adjacent layers (Hornik, et al., 1989).

A feedforward network computation with these backpropagation neural networks proceeds as follows:

1) The units in the input layer receive their activations in the form of an input pattern and this initiates the feedforward process,

2) The processing units in each layer receive outputs from other units and perform the following computations:

a) Compute their net input N_j,

$$N_j = \sum_{k=1}^{M} w_{jk}\ o_k$$

o_k = output from units impinging on unit j,
M = number of units impinging on unit j.

b) Compute their activation values from their net input values,

$$a_j = F_j(N_j)$$

F_j is usually a sigmoidal function.

c) Compute their outputs from their activation values. In the neural network type used in this study the output is the same as the activation value.

$$o_j = a_j$$

3) The output values are sent to other processing units along the outgoing connections.

4) This process continues until the processing units in the output layer compute their activation values. These activation values are the output of the neural computations.

The modification of the strengths of the connections in the Generalized Delta Rule, described in (Rumelhart, et al., 1986), is accomplished through the gradient descent on the total error in a given training case.

$$\Delta w_{ij} = \eta\ \delta_j\ o_j$$

In this equation, η = a learning constant called the "learning rate" and δ_j = gradient of the total error with respect to the net input at unit j. At the output units, δ_j is determined from the difference between the expected activations t_j and the computed activations a_j:

$$\delta_j = (t_j - a_j)\ F'(N_j)$$

where F' is the derivative of the activation function.

At the hidden units the expected activations are not known a priori. The following equation gives a reasonable estimate of δ_j for the hidden units:

$$\delta_j = (\sum_{k=1}^{M} \delta_k\ w_{jk})\ F'(N_j)$$

In this equation, the error attributed to a hidden unit depends on the error of the units it influences. The amount of error from these units attributed to the hidden unit depends on the strength of connection from the hidden unit to those units; a hidden unit with a strong excitatory connection to a unit exhibiting error will be "blamed" for this error, causing this connection strength to be reduced.

Up to now, backpropagation neural networks have been utilized in most neurocomputing applications due to their robustness in learning. Problems suitable for using backpropagation networks usually have the following features: 1) Certain relationships exist between the input and output variables, 2) A comprehensive set of data from tests or experiments is available, and 3) The knowledge to capture is included in the experimental data. A general architecture for backpropagation neural networks is shown in Figure 2.2.

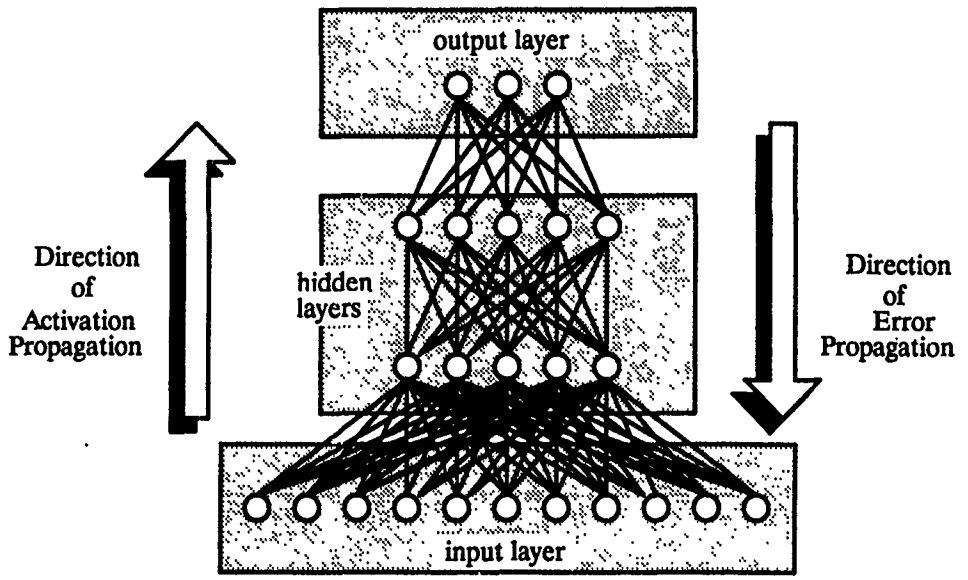


Figure 2.2 – A Sample Backpropagation Neural Network

From our discussion on the mechanism of backpropagation network, the main tasks involved in using the network are: 1) determination of architecture and 2) learning algorithms for the training of architectures. Though trial and error might have worked for certain simple problems in architecture determination, new and adaptive schemes are required in order to tackle real world complex problems with efficiency and elegance. In the following paragraph, some of the new approaches which in one way or the other made some improvement on the standard backpropagation algorithm are succinctly described.

References

1. Becker, S., and Le Cun, Y., "Improving the Convergence of Backpropagation Learning with Second Order Methods," *Proceedings of the 1988 Connectionist Models Summer School*, Carnegie Mellon University, Pittsburgh, 1988.

2. Gallant, A. R., and White, H., "There Exits A Neural Network that Does Not Make Avoidable Mistakes," *IEEE International Conference on Neural Networks*, Vol. I, pp. 657-664, 1988.

3. Hecht-Nielsen, R., "Kolmogorov's Mapping Neural Network Existence Theorem," *IEEE International Conference on Neural Networks*, Vol. III, pp. 11–14, 1987.

4. Hecht-Nielsen, R., "Theory of the Backpropagation Neural Networks," *Proceedings of the IEEE International Conf. on Neural Networks*, Vol. I, pp. 593-605, 1988.

5. Hornik, K., Stinchcomebe, M., and White, H., "Multilayer Feedforward Networks are Universal Approximators," *Neural Networks*, Vol. 2, pp. 359–366, 1989.

6. Le Cun, Y., "A Theoretical Framework for Backpropagation," *Proceedings of the 1988 Connectionist Models Summer School*, Carnegie Melon University, Pittsburgh, 1988.

7. Parker, D. B., "Learning Logic," *Invention report, S81–64, File 1*, Office of Technology Licensing, Standford University, October 1982.

8. Parker, D. B., "Optimal Algorithm for Adaptive Networks: Second Order Backpropagation, Second Order Direct Propagation, and Second Order Hebbian Learning," *Proceedings of the IEEE International Conference on Neural Networks*, Vol. II, pp. 168–176, 1987.

9. Rumelhart, D. E., Hinton, G. E., and Williams, R. J., "Learning Internal Representations by Error Propagation," in *Parallel Distributed Processing, Vol. 1: Foundations*, D. E. Rumelhart, and J. L., McClelland, (Eds.), The MIT Press, MA, 1986.

10. Warbos, P., *Beyond Regression: New Tools for Prediction and Analysis in Behavioral Sciences*, Ph.D. thesis, Harvard University, 1974.

11. Watrous, R. L., "Learning Algorithm For Connectionist Networks: Applied Gradient Methods Of Nonlinear Optimization," *Proceedings of the IEEE International Conference on Neural Networks*, Vol. II, pp. 619-627, 1987.

12. Windrow, B., Winter, R. G., and Baxter, R. A., "Learning Phenomena In Layered Neural Networks," *Proceedings of the IEEE International Conference on Neural Networks*, Vol. II, pp. 411-429, 1987.

2.2.3 Gram-Schmidt Backpropagation Network

It is realized that the training of a backpropagation network will be much easier and the convergence rate will be faster if the input vectors are least correlated, that is, orthogonal or independent. Based on this observation, Orfanidis (1990) proposed the Gram-Schmidt Neural Nets by inserting a Gram-Schmidt preprocessor at each layer in a regular backpropagation network. To store the preprocessed input vector impinging at each layer, an additional vector $\mathbf{Z}^n$ is required. The general architecture of the Gram-Schmidt network is shown is Fig. 2.3.

The decorrelation process on the input vector **X** using the Gram-Schmidt algorithm proceeds as follows: for i = 1, 2, . . ., M (number of units in a layer)

$$z_i = x_i - \sum_{j=1}^{i-1} b_{ij}\, z_j$$

or in matrix form, $\mathbf{X} = \mathbf{B}\,\mathbf{Z}$, where **B** is a unit lower triangular matrix.

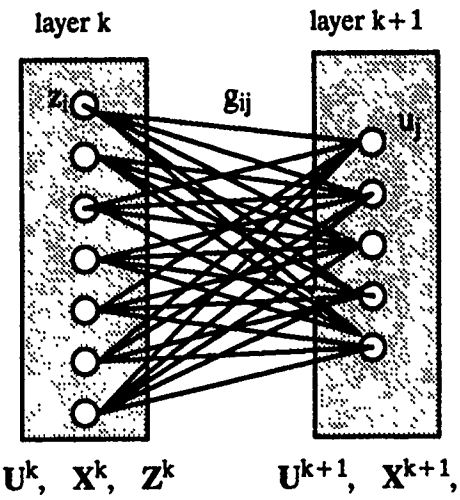
The training algorithm in the standard backpropagation mode is as follows:

1) Initialize the **G** weight matrix with random value and calculate the initial **B** matrix.

2) Feedforward computation: for k= 1, 2, . . . , N (# of layers)

   2.1) Solve $\mathbf{B}^k\,\mathbf{Z}^k = \mathbf{X}^k$ for $\mathbf{Z}^k$ through forward substitution,
   2.2) Calculate $\mathbf{U}^{k+1} = \mathbf{G}^k\mathbf{Z}^k$ and $\mathbf{X}^{k+1} = \mathbf{f}(\mathbf{U}^{k+1})$, where **f** is a vector containing sigmoid function evaluated with $\mathbf{U}^{k+1}$.

3) Error Calculation



Here U^k, X^k denote the input and output vector at each layer, and Z^k denotes the Gram-Schmidt preprocessed vector at layer k.

Figure 2.3 – A Sample Two Layers in a Gram-Schmidt Network

3.1) At the output layer: $e^N = D^N (d - X^N)$, where $D^N = \text{diag}\{ f'(u^N) \}$.

3.2) At the hidden layers: for k= N-1, N-2, . . . , 2

Solve $B^{kT} t^k = G^{kT} e^{k+1}$ for t^k through backward substitution and calculate $e^k = D^k t^k$.

4) Weight Update

$\Delta b_{ij}^k = \mu z_i^k z_j^k$ and $\Delta g_{ij}^k = \mu e_i^{k+1} z_j^k$.

It should be noticed that the effectiveness of using the Gram-Schmidt nets depends on the eigenvalue spread of the covariance matrix **R** of the input pattern X^0, where **R** is defined as:

$$\mathbf{R} = \sum_{\text{patterns}} \mathbf{X}^0 \mathbf{X}^{0T}.$$

If the ratio of the largest eigenvalue to the smallest eigenvalue of matrix **R** is large, the Gram-Schmidt preprocessor will be very effective.

References

1. Golub, G. H., and Van Loan, C. F., *Matrix Computations*, The Johns Hopkins University Press, Baltimore, Maryland, 1983.
2. Orfanidis, S. J., "Gram-Schmidt Neural Nets," *Neural Computation* 2, pp. 116–126, 1990.
3. Orfanidis, S. J., *Optimum Signal Processing*, 2nd Edition, McGrow-Hill Book Co., New York, 1988.

2.2.4 The Cascade-Correlation Neural Network

As have been pointed out by Falman and Lebiere (1990), main problems associated with convergence rate of backpropagation networks are the *step-size problem* and the *moving target problem*. The former refers to the use of constant step size and the latter to the inability of hidden nodes in quickly capturing regularity from only input signals and error signals without taking account of the lateral interactions, when the error surface changes frequently. Those problems are manifested by the determination of network architecture, especially the determination of the number of nodes in the hidden layers, which usually cannot be defined a priori. The Cascade-Correlation dynamic node generation network (Fahlman and Lebiere, 1990) provides some rational thinking in solving the learning problem.

A Cascade-Correlation Network starts with a basic network, then trains and adds new hidden units one by one, creating a multilayer structure. There are two processes involved in the construction of a Cascade-Correlation network. The first deals with the architecture generation and the second deals with the learning algorithm. According to Fahlman's description, hidden nodes are added to the network one by one. Each hidden node receives a connection from each of the network's original inputs and also from every pre-existing hidden units. Once a unit is added to the network, its weight on the input side is frozen and only the weight on the output side is trained. The unit creation algorithm essentially performs following operations: firstly it starts with a candidate unit (or a pool of candidate units) that receives connections from the network external inputs and from previous hidden units, with

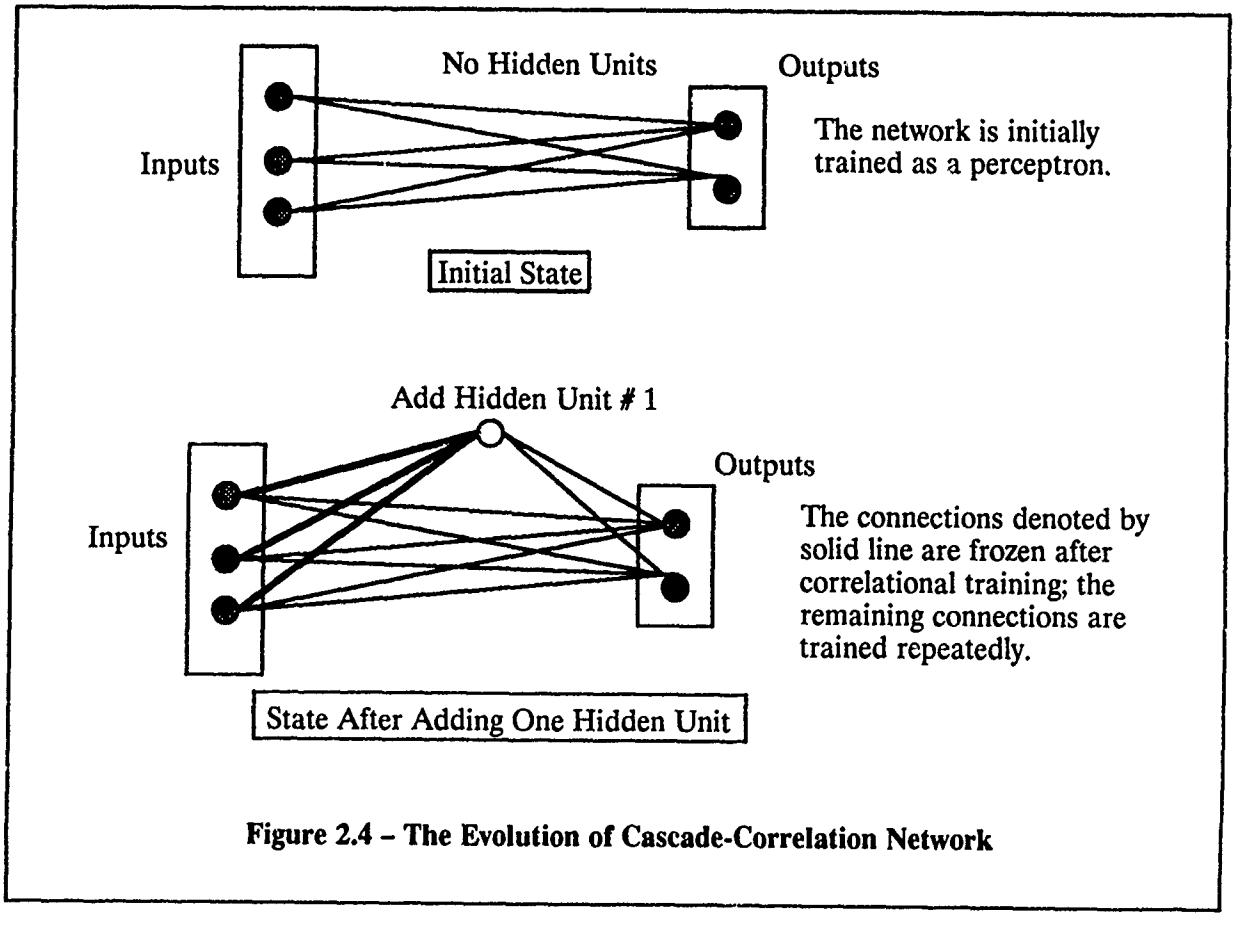


Figure 2.4 – The Evolution of Cascade-Correlation Network

the output of the candidate unit being disconnected to the trained network. Then the newly created candidate network is trained by using quickprop (Fahlman, 1988) on the training sets and only the candidate unit's input weights are adjusted after each pass. The training criterion for input weights is to maximize the correlation measurement S defined as

$$S = \sum_o \left| \sum_p (V_p - \bar{V})(E_{p,o} - \bar{E_o}) \right|,$$

which is the sum over all output units o of the magnitude of the correlation between V, the candidate unit's activation value, and E_0, the residual output error observed at unit o. When S stops improving, the new candidate is installed to the trained network and its input weights are frozen subsequently. This process continues until the concept in the training sets is properly captured by the network. The cascade architecture evolution process is illustrated in Fig. 2.4.

The main advantage of the Cascade-Correlation Architecture over other existing learning algorithms is that it learns very quickly and it systematically determines its network structure during the training process. The power of this learning algorithm has been illustrated through modeling the two spirals problem (Lang and Witbrock, 1988).

References

1. Fahlman S. E., and Lebiere, C., "The Cascade-Correlation Learning Architecture," *Technical Report CMU-CS-90-100*, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213, Feb. 1990.

2. Fahlman S. E., and Lebiere, C., "The Cascade-Correlation Learning Architecture," in *Advances in Neural Information Processing Systems 2*, D. S. Touretzky (Ed.), Morgan Kaufmann Publishers, Inc., pp. 524–532, 1990.

3. Fahlman, S. E., "Faster-Learning Variations on Backpropagation: An Empirical Study," in *Proceedings of the 1988 Connectionist Models Summer School*, Morgan Kaufmann Publishers, Inc., 1988.

4. Jacobs, R. A., "Increased Rate of Convergence Through Learning Rate Adaptation," *Technical Report* (COINS TR 87-117), Dept of Computer and Information Science, University of Massachusetts at Amherst, MA, 1987.

5. Lang, K. J., and Witbrock, M. J., "Learning to Tell Two Spirals Apart," in *Proceedings of the 1988 Connectionist Models Summer School*, Morgan Kaufmann, 1988.

6. Moody, J., "Fast Learning in Multi-resolution Hierarchies," in D. S. Touretzky (Ed.), *Advances in Neural Information Processing Systems 1*, Morgan Kaufmann, 1989.

2.2.5 The Self-Organizing Neural Network (SONN)

The SONN proposed by Tenorio and Lee (1989) is a supervised learning algorithm for architecture construction and refinement in feedforward neural networks. The learning process is controlled by a

modified Minimum Description Length criterion (Rissanen, 1983) which is also used as an optimality criterion to guide the construction of the network structure, instead of the simplistic mean-square error. The search for the correct model structure or network architecture is accomplished via Simulated Annealing (Kirkpatrick, et al., 1983) so that the node accepting rule varies at run-time according to a cooling temperature schedule.

It should be pointed out that SONN is proposed to solve a general system identification problem such that its structure bears some close relation to the representation of nonlinear systems. The SONN algorithm can be characterized by three components: 1) a generating rule of the primitive neuron transfer functions, 2) an evaluation method to assess the quality of the model, and 3) a structure search strategy via Simulated Annealing. The algorithm can be conceptually put in the following form:

1) Initialize the cooling temperature and the state with basic nodes,
2) Repeat the following procedures until the magnitude of the temperature is smaller than the terminal temperature for simulated annealing:

    2.1) Repeat the following computations until the number of new neurons is greater than the number of observations:
        2.1.1) Use the neuron generating rule to produce new neurons to the structure and calculate the energy corresponding to the current and new states;
        2.1.2) If the energy of the new state is smaller than that of the current state, then accept the neuron, else accept the neuron with a probability.
    2.2) Decrease the temperature via geometrical annealing sequence.

The performance of the algorithm was illustrated through modeling a highly chaotic time series for system identification and short time prediction and through comparison with standard backpropagation networks. Though the search algorithm with Simulated Annealing is a non-deterministic scheme, the SONN shows remarkable advantage over the standard backpropagation learning algorithm. The SONN requires far less samples to acquire an estimation of the system and the structure of the network is determined at run-time. Because of the use of Simulated Annealing in the learning algorithm, it is less susceptible to the problem with local minima and the convergence of learning does not depend on conditions of the initial set of weight. On the other hand, a system modeled with SONN algorithm shows better performance in prediction. However, three types of nodes with different functionalities are defined in this network. Needless to say, this algorithm has more parameters and is more complex than the straightforward backpropagation network. Certain fine tuning routines are needed for a successful implementation of the algorithm.

References

1. Ivakhnenko, A. G., "Polynomial Theory of Complex Systems," *IEEE Transactions on System, Man and Cybernetics*, SMC-1, No. 4, pp. 364-378, 1971.

2. Ikeda, S., Ochiai, M., and Sawarogi, Y., "Sequential GMDH Algorithm and Its Application to River Flow Prediction," *IEEE Transactions on System, Man and Cybernetics*, SMC-6, No. 7, pp. 473-479, July 1971.

3. Kassebaum, J., Tenorio, M. F., and Schaefers, C., "The Cocktail Party Problem: Speech/Data Signal Separation Comparison between Backpropagation and SONN," in *Advances in Neural Information Processing Systems II*, D. S. Touretzky (Ed.), Morgan Kaufmann Publishers, Inc., pp. 542-549, 1990.

4. Kirkpatrick, S., Gelatt, C. D., and Vecchi, M. P., "Optimization by Simulated Annealing," *Science*, Vol. 220, pp. 671–680, May 1983.

5. Rissanen, J., "A Universal Prior for Integers and Estimation by Minimum Length Criterion," *Annuals of Statistics*, Vol. 11, No. 2, pp. 416–431, 1983.

6. Tenorio, M. F. M., and Lee, W.-T., "Self-Organizing Neural Network for the Identification Problem," in *Advances in Neural Information Processing Systems 1*, D. S. Touretzky (Ed.), Morgan Kaufmann Publishers, Inc., pp. 57–64, 1989.

7. Tenorio, M. F. M., and Lee, W.-T., "Self-Organizing Neural Network for Optimum Supervised Learning," *TR-EE-89-30*, School of Electrical Engineering, Purdue University, June 1989.

2.2.6 The GrowNet Algorithm

The GrowNet algorithm is proposed by Smith (1990) to solve the topology determination problem associated with backpropagation networks. This approach bears some resemblance to the SONN algorithm described in the previous subsection in that it uses a heuristic rule to choose the next step in growing a net and each step is reversible; whereas the SONN algorithm adopts a stochastic search algorithm to generate suitable topology.

The GrowNet algorithm begins with a simple network and tries to minimize the simplistic mean-square error using gradient descent method. At each epoch, the prospect of further reduction on the error is checked. If the prospect is not desirable, then the net is enlarged by a growth process, otherwise the gradient descent continues. There are two components in the GrowNet process: a statistics gathering epoch, and the growing of a node. The statistics gathering epoch includes the computation of statistics on the correlation between the error of each node and the activation of other nodes, the estimation on benefit of growth at each node, and the determination of the node that will offer the greatest benefit after growing. Growing a node involves replacing the node with a more complex node or a group of nodes. Similar to the SONN algorithm, three types of nodes with different activation functions are also used in this network. The pseudocode as given in the report is as follows:

```
1) Declarations:
   flag gatherstats;

2) Initialization:
   Create net with a simple topology;
   Unset gatherstats;

2) REPEAT
     For (each exemplar) Do
           Collect derivatives of net parameters;
```

```
        If (gatherstats is set) Then
          Collect correlation statistics;
        End If;
    End For;
        If (task solved) Then
          Exit REPEAT;
        Elseif (gatherstats is set) Then
          Select most promising node;
          Grow most promising node;
          Imset gatherstats;
        Elseif (no further reduction of error likely) then
          set gatherstats;
        Else
          Update net parameters;
        End If;
    End REPEAT.
```

The characteristics of different parameters in the tuning of this algorithm has been studied and its performance on the Or-NxM tasks shows its promise in the improvement on the standard gradient descent method. However, this algorithm is not extensively tested so that its power and shortcomings are not well exposed yet. The best feature of this algorithm lies in its run-time determination of the network configuration or architecture, its moderate computational cost, and its compatibility with the standard backpropagation algorithm.

It is interesting to notice that SONN and GrowNet algorithms are seldom used by other researchers in the neural network community. One primary reason is that these two algorithms utilize three types of nodes or computational units to build a network, which in turn introduces more complexity and uncertainty in the implementation of these algorithms. Nevertheless, it is felt that these two algorithms are not well explored and fully understood yet. There is still much work to be done in this direction.

References

1. Mozer, M. C., and Smolensky, P., "Using Relevance to Reduce Network Size Automatically," *Connection Science*, Vol. 1, No. 1, pp. 3–16, 1989.
2. Sietsma, J., and Dow, R. J. F., "Neural Net Pruning – Why and How," in *Proceedings of 1988 IEEE International Conference on Neural Networks*, San Diego, CA, 1988.
3. Smith, G., "Backpropagation with Dynamic Topology and Simple Activation Functions," *Technical Report (TR 90-12)*, Discipline of Computer Science, School of Information Science and Technology, Flinders University of South Australia, PO Box 2100, Adelaide, SA 5001, 1990.

2.3 Higher Order Schemes for Feedforward Networks

As has been stated before, the Generalized Delta Rule in backpropagation networks performs a gradient descent search in the weight space for the minimization of a mean-squared error function.

Therefore, it is easy to postulate that all the minimization schemes are applicable to the learning algorithm derivation for multilayer feedforward networks. From numerical analysis, the (steepest) gradient descent method is a first order scheme and has poor numerical property in terms of convergence rate and the ability to handle ill-conditioning of the system. Higher order methods such as Newton's methods and Quasi-Newton methods by including information on the second order derivatives, that is, the Hessian matrix of the system, have far better numerical properties than the steepest descent method. However, the computation expense involved with the determination of higher order information is very expensive and it also requires more storage space. On the other hand, the Conjugate Gradient Method or Preconditioned Conjugate Gradient Methods would provide faster learning algorithms because of their superlinear rate of convergence and the saving in storage space (Hageman and Young, 1981; Golub and VanLoan, 1983).

Nevertheless, before a higher order scheme is considered as a legitimate candidate in the realm of learning algorithms, it should be derived in a form which is computationally efficient and suitable for local implementation. It should also conserve the intrinsic parallelism of operations of the network. In a backpropagation network, the formula for weight update is:

$$\Delta w(t) = -\epsilon \, \partial E/\partial w(t) + \alpha \Delta w(t-1)$$

where ϵ is the learning rate and α the momentum factor. The update of weights proceeds either in batch mode or in on-line mode. The former refers to the update of weights only after all the training sets have been presented to the network, and the later refer to the update of weights after presenting each training set. For second and higher order algorithms, backpropagation is usually implemented in the batch mode. It thus becomes obvious that any improvement on the learning algorithm alone should involve the adaptive determination of the two learning parameters (ϵ and α), whereas these two parameters are set as constants in the standard backpropagation network. To date, numerous schemes have been proposed to improve the learning mechanisms in backpropagation networks by incorporating higher order information of the system or using heuristic rules to guide the adaptation of the learning parameters. In the following paragraphs, some of the new schemes are sketchily described.

References

1. Golub, G. H., and VanLoan, C. F., *Matrix Computations*, The Johns Hopkins University Press, 1983.

2. Hageman, L. A., and Young, D. M., *Applied Iterative Methods*, Academic Press, New York, 1981.

2.3.1 Quickprop

The Quickprop algorithm was proposed by Falhman (1988) to improve the rate of convergence of the backpropagation network through adaptive calculation of the momentum factor α. It is a second order method in a sense, based loosely on Newton's method, but it is more heuristic than formal. The information required is the gradient of the previous training epoch and that of the current, along with

the difference between that of the previous and current. Therefore, the weight update formula is:

$$\Delta w(t) = -\epsilon \, \partial E/\partial w(t) + \frac{\partial E/\partial w(t)}{\partial E/\partial w(t-1) - \partial E/\partial w(t)} \Delta w(t-1)$$

According to Falhman, the Quickprop algorithm is derived based on two crude assumptions: 1) the error vs. weight curve for each weight can be approximated by a parabola, and 2) the change in the slope of the error curve, as seen by each weight, is independent of the other weights that are changing at the same time. Though those are simple assumptions, the resulting algorithm gives substantial improvement on the convergence rate over the standard scheme when tested on the Encode/Decode tasks. The speedup over standard backpropagation algorithm is about one order of magnitude (10 times) on training a small set of benchmark problems, and the algorithm seems scaled-up well.

Reference

1. Fahlman, S., "Faster-Learning Variations on Backpropagation: An Empirical Study," *Proceedings of the 1988 Connectionist Models, Summer School*, D. Touretzky, G. Hinton and T. Sejnowski, (Eds.), pp. 38–51, Carnegie Mellon University, Morgan Kaufman Publishers, Inc., 1989.

2.3.2 Quasi-Newton Style Methods

Direct use of the Newton Method to neural network is inappropriate for the following reasons. At first, Newton method gains quadratic or nearly quadratic convergence only when the starting point in the solution space is within a convex region of the function and the size of that neighborhood diminishes with increasing number of variables, whereas a randomized initial weight matrix is routinely used in training a neural network. Secondly, the computation of the Hessian matrix and its inverse requires expensive computation and intensive storage (Dennis and More, 1977). Though a Quasi-Newton method such as the BFGS (Broyden–Fletcher–Goldfab–Shanno) algorithm (Dennis and More, 1977; Dennis and Schnabel, 1983) has better numerical performance, especially on the storage usage, than the Newton method, like the Newton method it also uses global information for the updating of weights such that its use as an efficient learning rule for large problems in practice is questionable. In this subsection, the BFGS method is briefly described to show the flavor of Quasi-Newton methods. Some other schemes with a connection to the use of information on second order derivatives are thus classified as Quasi-Newton style methods.

2.3.2.1 BFGS Method

The use of the BFGS method as a learning algorithm for feedforward networks has been investigated by Watrous (1987). In nonlinear minimization, if the objective function or the error function is approximated as a quadratic function through Taylor's expansion, then

$$E(w + \Delta w) \approx E(w) + g^T \Delta w + \frac{1}{2} \Delta w^T G \Delta w$$

where g is the gradient vector defined as $g = \nabla E(w)$, and G is the Hessian matrix. In the Newton Method, the minimum can be directly computed by solving the system of equations, namely:

$$\Delta w = G^{-1} g$$

In a Quasi-Newton method, instead of calculating the Hessian matrix and its inverse or solving a system of equations, the inverse matrix of the Hessian is approximated iteratively with H. The basic quasi-Newton algorithm consists of the following steps (Dennis and Schnabel, 1983):

1) Calculate a search direction $s = - H g$;

2) Perform line search in the s direction, that is, minimize E(w) along s;

3) Update H using different schemes such as the BFGS algorithm.

The difference among Quasi-Newton methods lies in the utilization of different updating schemes for H. Nevertheless, the BFGS Hessian update is symmetric and positive definite, making the algorithm numerically more stable than other schemes. If we define $\gamma_i = g_i - g_{i-1}$, and $\delta_i = w_i - w_{i-1}$, then the BFGS update is of the following form (Dennis and More, 1977),

$$H_i = H_{i-1} + (1 + \frac{\gamma^T H \gamma}{\delta^T \gamma}) \frac{\delta \delta^T}{\delta^T \gamma} - \frac{\delta \gamma^T H + H \gamma \delta^T}{\delta^T \gamma}$$

In Watrous' study, the performance of BFGS algorithm is compared with that of backpropagation on the training of the XOR problem and a small multiplexor problem. It was reported that the BFGS method converged in significantly fewer iterations and had a better error tolerant property. However, each BFGS iteration still requires $O(n^2)$ operations (Dennis and More, 1977), compared to O(n) for backpropagation. On the other hand, because the method has not been extensively tested on different problems, the robustness of the BFGS method is not well understood yet.

2.3.2.2 The Pseudo-Newton Algorithm

The pseudo–Newton algorithm was proposed by Becker and le Cun (1988) to approximate the information of the second order derivatives and to include it in the learning algorithm. The algorithm only calculates the diagonal terms of the Hessian matrix and ignoring the off-diagonal terms. From intuition on numerical analysis, the algorithm would work very well for diagonally dominant systems. By using the absolute value of the diagonal Hessian terms, the pseudo–Newton step for weight update is defined in the following form,

$$\Delta w = - \frac{\partial E(w)/\partial w}{|\partial^2 E(w)/\partial^2 w + \mu|}$$

Where μ is a small value to improve the conditioning of the Hessian when in regions of very small curvature such as at inflection points and plateaus.

The performance of this algorithm has been tested on the encode/decode problem and appears to have a slightly faster convergence rate. However, it is reported that if the initial weights are set to be very large or very small values, the algorithm fails to converge. Besides, in some regions where the

gradient is very steep and the curvature is very shallow, the algorithm tends to compute steps that are too large. The problem might rest on the lack of line search routine with the use of second order information. This indicates that care should be exercised in using higher order methods to the learning problem because the system itself is more statistically bound.

2.3.2.3 The Optimal Second Order Methods (OSOM)

Parker (1987) derived his optimal second order methods by using an efficient approximation to the Newton Method in the calculation of the second derivatives. Instead of minimizing the absolute squared error of the system, it tries to minimize the average squared error which is expressed as exponentially weighted average with time constant μ. If the total squared error is represented as $\epsilon^T\epsilon$, the the average squared error is defined as

$$\overline{\epsilon^T\epsilon} = \mu \int_{-\infty}^{t} \epsilon^T\epsilon \; e^{-\mu[t-\tau]} \; d\tau$$

where e is the natural logarithms.

The optimality condition can be derived in two steps. Initially, calculate the derivative of the average squared error with respect to the weights by holding t temporally constant and set the derivative to be zero. Then, define the optimality criterion in terms of the optimal path through weight space that the weights should follow as the network is trained. That is, to reactivate the time variable and calculate the derivative of that derived in the first step with respect to time, and let it be zero. The resulted differential equation is then the first order optimal algorithm. Through applying numerical treatment to the first order optimal algorithm, the second order algorithm is of the following form

$$\frac{\partial^2 w}{\partial^2 t} = -2\alpha \; [\mu \frac{\partial \epsilon^T}{\partial w}\epsilon + (\frac{\partial \epsilon^T}{\partial w}\frac{\partial \epsilon}{\partial w^T} + \frac{\partial^2 \epsilon^T}{\partial w \partial w^T}\epsilon) \; \frac{\partial w}{\partial t}]$$

It is easy to realize that this algorithm is not simple and the implementation scheme should be carefully derived. To date, the implementation scheme and its performance on numerical simulation has not been reported though the author claimed that it would be published. Even with this shortcoming, it is still worthwhile and insightful to look at the unique scheme in deriving the optimality criterion.

References

1. Becker, S., and le Cun, Y., "Improving the Convergence of Backpropagation Learning with Second Order Methods," *Proceedings of the 1988 Connectionist Models, Summer School*, D. Touretzky, G. Hinton and T. Sejnowski (Eds.), pp. 29–37, Carnegie Mellon University, Morgan Kaufman Publishers, Inc., 1989.

2. Dennis Jr., J. E., and More, J., "Quasi-Newton Methods, Motivation and Theory," *SIAM Review*, Vol. 19, No. 1, pp. 46–89, 1977.

3. Dennis Jr., J. E., and Schnabel, R., *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Printice-Hall, Englewood Cliffs, New Jersey, 1983.

4. Hageman, L. A., and Young, D. M., *Applied Iterative Methods*, Academic Press, New York, 1981.

5. Le Cun, Y., "A Theoretical Framework for Backpropagation," *Proceedings of the 1988 Connectionist Models, Summer School*, D. Touretzky, G. Hinton and T. Sejnowski (Eds.), pp. 21-28, Carnegie Mellon University, Morgan Kaufman Publishers, Inc., 1989.

6. Parker, D. P., "Optimal Algorithms for Adaptive Networks: Second Order Back Propagation, Second Order Direct Propagation, And Second Order Hebbian Learning," *Proceedings of the First IEEE International Conference on Neural Networks*, II-593, 1987.

7. Watrous, R. L., "Learning Algorithm For Connectionist Networks: Applied Gradient Methods Of Nonlinear Optimization," *Proceedings of the IEEE International Conference on Neural Networks*, II-619, 1987.

2.3.3 The Delta-Bar-Delta Algorithm (DBD)

It has been observed through previous description on higher order schemes that incorporating information on second order derivatives does not necessarily guarantee an improvement over the backpropagation learning algorithm. Experience shows that sometimes insightful heuristics would be beneficial if they are properly appraised. Jacobs (1988) proposed the Delta-Bar-Delta algorithm for adapting the learning rate with consideration to local gradient information. Inspired by Kesten's work (1958) on the steepest descent method that the weight value is oscillating if consecutive changes of a weight possess opposite signs, Saridis (1970) uses this observation to increase and decrease the learning rate in the following way: increasing the learning rate if consecutive derivatives of a weight possess the same sign, and decreasing the learning rate otherwise.

The Delta-Bar-Delta algorithm is derived based on the following heuristic rules: 1) Each weight has its own learning rate, 2) Every learning rate should be allowed to vary over time, 3) Increase the learning rate for a parameter if the derivative of the parameter possesses the same sign for several consecutive time steps, and 4) Decrease the learning rate for a parameter if the derivative of the parameter flips signs for several consecutive time steps. Based on these heuristic rules, the scheme for modifying the learning rate is defined as follows:

$$\Delta\epsilon(t) = \begin{cases} \kappa & \text{if } \bar{g}(t-1)g(t) > 0 \\ -\phi\epsilon(t) & \text{if } \bar{g}(t-1)g(t) < 0 \\ 0 & \text{otherwise} \end{cases}$$

where $g(t) = \nabla E(w(t))$ and $\bar{g}(t) = (1-\theta)\,g(t) + \theta\,\bar{g}(t-1)$. The parameters of κ, ϕ and θ are specified by the user.

The performance of the algorithm has been studied in training of the Quadratic surfaces task, the XOR problem, the Multiplexer problem and the Binary-to-Local problem and the speed-up ranges from 2 times to 2 orders of magnitude depending on the nature of the problem. The problem with this

algorithm is that the learning rate sometimes goes wild even with a small κ, and the tuning of ϕ is hard and sometimes contradictory.

2.3.3.1 The Extended Delta-Bar-Delta Algorithm (EDBD)

The EDBD algorithm is designed by Minai and Williams (1990) to improve the performance of the DBD algorithm with following modifications: 1) The learning rate increase is made an exponentially decreasing function of $|\bar{g}(t)|$ instead of constant κ; 2) Including the momentum part in the learning algorithm and letting the momentum vary with time; 3) A ceiling is defined for both the learning rate and momentum parameter; and 4) Memory and recovery are incorporated into the algorithm. After considering these modifications, the EDBD algorithm is of the following form:

$$\Delta w_{ij}(t) = -\eta_{ij}(t)\ \partial E/\partial w_{ij}(t) + \mu_{ij}\Delta w_{ij}(t-1)$$

$$\eta_{ij}(t+1) = \mathrm{Min}[\eta_{max},\ \eta_{ij}(t) + \Delta\eta_{ij}(t)]$$

$$\mu_{ij}(t+1) = \mathrm{Min}[\mu_{max},\ \mu_{ij}(t) + \Delta\mu_{ij}(t)]$$

$$\Delta\eta_{ij}(t) = \begin{cases} \kappa_l \exp(-\gamma_l|\overline{\delta_{ij}}(t)|) \\ -\phi_l\eta_{ij}(t) \\ 0 \end{cases}$$

$$\Delta\mu_{ij}(t) = \begin{cases} \kappa_m \exp(-\gamma_m|\overline{\delta_{ij}}(t)|) & \text{if } \overline{\delta_{ij}}(t-1)\delta_{ij}(t) > 0 \\ -\phi_m\eta_{ij}(t) & \text{if } \overline{\delta_{ij}}(t-1)\delta_{ij}(t) < 0 \\ 0 & \text{otherwise} \end{cases}$$

where $\delta_{ij}(t) = \nabla E(w_{ij}(t))$ and $\overline{\delta_{ij}}(t) = (1-\theta)\,\delta_{ij}(t) + \theta\,\overline{\delta_{ij}}(t-1)$. The parameters of κ_l, ϕ_l, η_{max}, γ_l and κ_m, ϕ_m, μ_{max}, θ, γ_m are specified by the user.

The performance of the EDBD algorithm is studied on the simulation of the XOR problem and the quadratic function problem and it shows that the EDBD algorithm has converged in all the cases. Besides, it is a more robust scheme than the DBD algorithm.

References

1. Jacobs, R. A., "Increased Rates of Convergence Through Learning Rate Adaptation," *Neural Network*, Vol. 1, pp. 295–307, 1988.
2. Kesten, H., "Accelerated Stochastic Approximation," *Annals of Mathematical Statistics*, 29, pp. 41–59, 1958.
3. Minai, A. A., and Williams, R. D., "Acceleration of Backpropagation through Learning Rate and Momentum Adaptation," *Proceedings of the International Joint Conference on Neural Networks*, Washington, D. C., I–676, January 1990.
4. Saridis, G. N., "Learning Applied to Successive Approximation Algorithms," *IEEE Transactions on Systems Science and Cybernetics*, SSC-6, pp. 97–103, 1970.

2.3.4 Adaptive Stepsize On-line Backpropagation

Backpropagation can be implemented in batch mode or in on-line mode during training on the data set. For second and higher order algorithms, only batch mode backpropagation is realized because of the nature of higher order methods. Using heuristics, Chen and Mars (1990) proposed an on-line mode backpropagation learning with stepsize adaptation. The algorithm is as follows:

$$\begin{aligned} \alpha(t) &= \alpha(t-1)\ (1-f(t)\sqrt{E(t)}\ ) \\ f(t) &= u_1 f(t-1) + u_2 \Delta E(t) \\ \Delta E(t) &= E(t) - E(t-1) \end{aligned}$$

where $\alpha(t)$ is the stepsize for the gradient term in the weight update formula, $\Delta E(t)$ is the decrement of $E(t)$ and $f(t)$ is a filtered version of $\Delta E(t)$. It can easily be seen that the equation for $f(t)$ represents a first order low-pass recursive filter. The parameters u_1 and u_2 are used to control the adaptation. For large u_1 and small u_2, the adaptation is slow but more stable, otherwise the adaptation is fast but may lead to oscillation. For the simulation problems, $u_1 = 0.9$ and $u_2 = 0.3$ have been used with success.

It has pointed out that this algorithm would work much better on complex problems than on simple problems because the adaptation process needs certain time to settle to be fully effective. The disadvantage of this algorithm is that it is not effective on the flat region because the algorithm makes the weights on the hidden units prematurely saturated. To overcome this problem, it is suggested that differential step size should be used such that the step size for weight updating between hidden and output layer is larger than that between the input and hidden layer, and usually the latter is about 0.1–0.5 of the former. Though performance of the algorithm is far better than the standard backpropagation algorithm, it is a bit slower than Quickprop proposed by Falhman (1988).

References

1. Chen, J. P., and Mars, P., "Stepsize Variation Methods for Accelerating the Backpropagation Algorithm," *Proceedings of the International Joint Conference on Neural Networks*, Washington, D. C., Jan. 15–19, 1990, I-601.

2. Fahlman, S., "Faster-Learning Variations on Backpropagation: An Empirical Study," *Proceedings of the 1988 Connectionist Models, Summer School*, D. Touretzky, G. Hinton and T. Sejnowski, (Eds.), pp. 38–51, Carnegie Mellon University, Morgan Kaufman Publishers, Inc., 1989.

2.3.5 Minkowski-r Backpropagation

It is almost routine that most connectionist learning models are implemented using a gradient descent in a least squares error function, that is, the error signals are Euclidian. People may then ask: how about deriving learning models based on non-Euclidian error measurement? Hanson and Burr (1988) answered this question with an elegant study on the backpropagation learning using Minkows-

ki-r power metrics as the error measurement. The derivation of the algorithm is similar to that of backpropagation by Rumelhart, Hinton and Williams (1986).

Using Minkowski-r power metrics, the error can be represented in the following general form:

$$E = \frac{1}{r}\sum_i |(y_i - \hat{y}_i|)^r$$

Then the gradient in the general Minkiwski-r case is

$$\frac{\partial E}{\partial w_{hi}} = (|y_i - \hat{y}_i|)^{r-1}\ y_i(1-y_i)y_h\ \mathrm{sgn}(y_i - \hat{y}_i)$$

and the weight update formula is thus

$$\Delta w_{hi}(n+1) = -\epsilon\ \partial E/\partial w_{hi} + \alpha \Delta w_{hi}(n)$$

The weight updating for the hidden layer proceeds in the same way as in the Euclidian case by simply substituting the Minkowski-r gradient.

Through numerical analysis, the behavior of learning model changes with the variation of the parameter r because changing the value of r basically results in a reweighting of errors from output bits. In one respect, varying the value of r may be useful for various aspects of representing information in the feature domain. For example, if the distribution of feature vectors is non-Gaussian, then the $r = 2$ case, that is, the Euclidian error case, will not be a maximum likelihood estimator of the weights. In fact, $r = 1$ would be right for modeling Laplacian type distribution. In general, when $r < 2$, it is recommended that $r = 1.5$ may be optimal for many noise reduction problems; when $r > 2$, it tends to weight large deviations such that simpler generalization surfaces may be created. However, it is observed that the convergence time tends to grow almost linearly with the increase of r. On the other hand, the implementation of the learning algorithm is more complex as the Minkowski-r gradient is nonlinear.

This approach is unique in that it looks at the same problem from a different perspective. Of course, further study is needed to explore the research and application potential in this direction.

References

1. Hanson, S. J., and Burr, D. J., "Minkowski-r Backpropagation: Learning in Connectionist Models with Non-Euclidian Error Signals," in *Neural Information Processing Systems*, D. Z. Anderson (Ed.), American Institute of Physics, New York, pp. 348–357, 1988.

2. Rumelhart, D. E., Hinton, G. E., and Williams, R., "Learning Internal Representation by Error Propagation," *Nature*, Vol. 323, pp. 533–536, 1986.

2.4 Other Approaches

In addition to the samples of algorithms described in previous paragraphs, there are a lot of other algorithms which improve the learning performance of standard backpropagation network in one way

or the other. The architectural determination of backpropagation network at run-time has also explored by other researchers (Bailey, 1990; Jockusch, 1990; Nevard, 1990). The learning models are also generalized to the complex plane to handle a special kind of problem (Clarke, 1990, Kim and Guest, 1990). New schemes have been proposed by incorporating stochastic training techniques (Day and Camporese, 1990; Kolen, 1988), using extrapolatory methods (Dewan and Sontag, 1990), including fuzzy theory (Fu, 1990; Oden, 1988), and other algebraic and numerical techniques. To illustrate the rich and fruitful research in this area, a selected reference listing is provided in this section. For details of each algorithm, the original article should be consulted rather than the short description provided in this section.

General References

1. Abe, S., "Learning by Parallel Forward Propagation," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17-21, 1990, III-99.

2. Akaho, S., and Amari, S., "On the Capacity of Three-Layer Networks," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17-21, 1990, III-1.

3. Allred, L. G., and Kelly, G. E., "Supervised Learning Techniques for Backpropagation Networks," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17-21, 1990, I-721.

4. Almeida, L. B., "A Learning Rule for Asynchronous Perceptrons with Feedback in A Combinatorial Environment," *Proceedings of the First IEEE International Conference on Neural Networks*, San Diego, 1987, II-609.

5. Baba, N., "A Hybrid Algorithm for Finding the Global Minimum of Error Function Neural Network," *Proceedings of the International Joint Conference on Neural Networks*, Washington, D. C., Jan. 15-19, 1990, I-585.

6. Bailey, A. W., "Automatic Evolution of Neural Net Architecture," *Proceedings of the International Joint Conference on Neural Networks*, Washington, D. C., Jan. 15-19, 1990, I-589.

7. Baldi, P., "Linear Learning: Landscapes and Algorithms," *Advances in Neural Information Processing Systems I*, D. S. Touretzky, Editor, Morgan Kaufmann Publishers, Inc., 1989, pp. 65.

8. Barto, A. G., and Jordan, M. I., "Gradient Following Without Backpropagation in Layered Networks," *Proceedings of the First IEEE International Conference on Neural Networks*, San Diego, 1987, II-629.

9. Battiti, R., "Optimization Methods for Backpropagation: Automatic Parameter Tuning and Faster Convergence," *Proceedings of the International Joint Conference on Neural Networks*, Washington, D. C., Jan. 15-19, 1990, I-593.

10. Becker, S., and le Cun, Y., "I. proving the Convergence of Backpropagation Learning with Second Order Methods," *Proceedings of the 1988 Connectionist Models, Summer School*, D. Touretzky, G. Hinton and T. Sejnowski (Eds.), pp. 29-37, Carnegie Mellon University, Morgan Kaufman Publishers, Inc., 1989.

11. Bochereau, L., Bourgine, P., "A Generalist-Specialist Paradigm for Multilayer Neural Networks," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17-21, 1990, III-87.

12. Bodenhausen, U., "The Tempo-Algorithm: Learning in a Neural Network with Variable Time Delays," *Proceedings of the International Joint Conference on Neural Networks*, Washington, D. C., Jan. 15-19, 1990, I-597.

13. Bodenhausen, U., "Learning Internal Representations of Pattern Sequences in A Neural Network with Adaptive Time-Delays," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17-21, 1990, III-113.

14. Brown, H. K., Lange, D. F., and Hart, J. L., "Orthogonal Extraction Training algorithms," *Proceedings of the International Joint Conference on Neural Networks*, Washington, D. C., Jan. 15-19, 1990, I-537.

15. Brodsky, S. A., and Guest, C.C., "Binary Backpropagation in Content Addressable Memory," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17-21, 1990, III-205.

16. Burrascano, P., and Lucci, P., "Smoothing Backpropagation Cost Function by Delta Constraining," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17-21, 1990, III-75.

17. Burrascano, P., and Lucci, P., "A Learning Rule in the Chebyshev Norm for Multilayer Perceptrons," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17-21, 1990, III-81.

18. Cater, J. P., "Successfully Using Peak Learning Rates of 10 (and Greater) in Backpropagation Networks with the Heuristic Learning Algorithm," *Proceedings of the First IEEE International Conference on Neural Networks*, San Diego, 1987, II-645.

19. Chen, J. P., and Mars, P., "Stepsize Variation Methods for Accelerating the Backpropagation Algorithm," *Proceedings of the International Joint Conference on Neural Networks*, Washington, D. C., Jan. 15-19, 1990, I-601.

20. Chen, M. S., and Manry, M. T., "Backpropagation Representation Theorem Using Power Series," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17-21, 1990, I-643.

21. Cheung, R. K. M., Lustig, I., and Kornhauser, A.L., "Relative Effectiveness of Training Set Patterns for Backpropagation," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17-21, 1990, I-673.

22. Cho, S. -B., and Kim, J. H., "An Accelerated Learning Method with Backpropagation," *Proceedings of the International Joint Conference on Neural Networks*, Washington, D. C., Jan. 15-19, 1990, I-605.

23. Clarke, T. L., "Generalization of Neural Networks to the Complex Plane," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17-21, 1990, II-435.

24. Day, D. S., "Towards Integrating Automatic and Controlled Problem Solving," *Proceedings of the First IEEE International Conference on Neural Networks*, San Diego, 1987, I-661.

25. Day, S. P., and Camporese, D. S., "Stochastic Training Technique for Feed-Forward Neural Networks," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17–21, 1990, I–607.

26. Dewan, H. M., and Sontag, E. D., "Extrapolatory Methods for Speeding Up the BP Algorithm," *Proceedings of the International Joint Conference on Neural Networks*, Washington, D. C., Jan. 15–19, 1990, I–613.

27. Fahlman, S., "Faster-Learning Variations on Backpropagation: An Empirical Study," *Proceedings of the 1988 Connectionist Models, Summer School*, D. Touretzky, G. Hinton and T. Sejnowski (Eds.), pp. 38–51, Carnegie Mellon University, Morgan Kaufman Publishers, Inc., 1989.

28. Fakhr, W., and Elmasry, M. I., "A Fast Learning Technique for the Multi-Layer Perceptron," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17–21, 1990, III–257.

29. Fausett, D. W., "Strictly Local Backpropagation," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17–21, 1990, III–125.

30. Fogel, D. B., Fogel, L. J., and Porto, V. W., "Evolutionary Programming for Training Neural Networks," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17–21, 1990, I–601.

31. Fu, L. M., "Backpropagation in Neural Networks with Fuzzy Conjunction Units," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17–21, 1990, I–613.

32. Gallant, S. I., and Smith, D., "Random Cells: An Idea Whose Time Has Come and Gone . . . and Come Again?" *Proceedings of the First IEEE International Conference on Neural Networks*, San Diego, 1987, II–671.

33. Gallant, A. R., and White, H. "There Exits A Neural Network That Does Not Make Avoidable Mistakes," *Proceedings of the IEEE International Conference on Neural Networks*, 1988, I–657.

34. Godfrey, K. R. L., and Attikiouzel, Y., "Polar Backpropagation," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17–21, 1990, III–143.

35. Goggin, S. D. D., Gustafson, K. E., and Johnson, K. M., "Connectionist Nonlinear Over-Relaxation," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17–21, 1990, III–179.

36. Guillerm, T. J., Cotter, N. E., "Neural Networks in Noisy Environment: A Simple Temporal Higher Order Learning for Feed-Forward Networks," *Proceedings of the IEEE International Conference on Neural Networks*, 1988, I–657.

37. Hagiwara, M., "Accelerated Backpropagation Using Unlearning Based on Hebb Rule," *Proceedings of the International Joint Conference on Neural Networks*, Washington, D. C., Jan. 15–19, 1990, I–617.

38. Hagiwara, M., "Novel Backpropagation Algorithm for Reduction of Hidden Units and Acceleration of convergence Using Artificial Selection," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17–21, 1990, I–625.

39. Hagashino, J., De Greef, B. L., and Persoon, E. H. J., "Numerical Analysis and Adaptation Method for Learning Rate of Backpropagation," *Proceedings of the International Joint Conference on Neural Networks*, Washington, D. C., Jan. 15–19, 1990, I–627.

40. Hanson, S. J., and Burr, D. J., "Minkowski-r Backpropagation: Learning in Connectionist Models with Non-Euclidian Error Signals," in *Neural Information Processing Systems*, D. Z. Anderson, Editor, American Institute of Physics, New York, pp. 348–357, 1988.

41. Hecht-Nielsen, R. "Kolmogorov's Mapping Neural Network Existence Theorem," *Proceedings of the IEEE International Conference on Neural Networks*, 1987, III–11.

42. Hechet-Nielsen, R., "Theory of the Backpropagation Neural Networks," *Proceedings of the IEEE International Conference on Neural Networks*, 1988, I–593.

43. Himmelblau, D. M., "Introducing Efficient Second Order Effects into Backpropagation Learning," *Proceedings of the International Joint Conference on Neural Networks*, Washington, D. C., Jan. 15–19, 1990, I–631.

44. Hornik, K., Stinchcombe, M., and White, H., "Multilayer Feedforward Network Are Universal Approximators," *Neural Networks*, Vol. 2, pp. 359-366, 1989.

45. Hu, C.– L. J., "A Novel, One Step, Geometrical, Supervised Learning Scheme," *Proceedings of the International Joint Conference on Neural Networks*, Washington, D. C., Jan. 15–19, 1990, I–635.

46. Hush, D. R., and Salas, J. M., "Improving the Learning Rate of Backpropagation with the Gradient Reuse Algorithm," *Proceedings of 1988 IEEE International Conference on Neural Networks*, I–441.

47. Irie, B., and Miyake, S., "Capabilities of Three–Layered Perceptrons," *Proceedings of the IEEE International Conference on Neural Networks*, 1988, I-641.

48. Izui, Y., and Pentland, A., "Speeding Up Backpropagation," *Proceedings of the International Joint Conference on Neural Networks*, Washington, D. C., Jan. 15–19, 1990, I–639.

49. Jockusch, S., "A Neural Network Which Adapts Its Structure to A Given Set of Patterns," in *Parallel Processing in Neural Systems and Computers*, R. Echmiller, G. Hartmann, G. Hauske, Editors, North-Holland, pp. 169, 1990.

50. Jones, K. L., Lustig, I. J., and Kornhauser, A. L., "Optimization Techniques Applied to Neural Networks: Line Search, Implementation for Backpropagation," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17–21, 1990, III–933.

51. Jones, R. D., Lee, Y. C., Barnes, C. W., Flake, G.W., Lee, K., Lewis, P. S., and Qian, S., "Function Approximation and Time Series Prediction with Neural Networks," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17–21, 1990, I–649.

52. Judd, S., "Learning in Networks is Hard," *Proceedings of the First IEEE International Conference on Neural Networks*, San Diego, 1987, II–685.

53. Kim, M. S., and Guest, C. C., "Modification of Backpropagation Networks for Complex-Valued Signal Processing in Frequency Domain," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17–21, 1990, III–27.

54. Kolen, J. F., "Faster Learning Through a Probabilistic Approximation Algorithm," *Proceedings of 1988 IEEE International Conference on Neural Networks*, I-449.

55. Kollias, S., and Anastassiou, D., "Adaptive Training of Multilayer Neural Networks Using a Least Squares Estimation Technique," *Proceedings of 1988 IEEE International Conference on Neural Networks*, I-383.

56. Kong, Y. H., and Noetzel, A. S., "The Piecewise Linear Neural Network: Training and Recognition," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17-21, 1990, III-245.

57. Kramer, A. H., and Sangiovanni-Vincentelli, A., "Efficient Parallel Learning Algorithms for Neural Networks," *Advances in Neural Information Processing Systems I*, D. S. Touretzky, Editor, Morgan Kaufmann Publishers, Inc., pp. 40, 1989.

58. Kruschke, J. K., "Creating Local and Distributed Bottlenecks in Hidden Layers of Backpropagation Networks," *Proceedings of the 1988 Connectionist Models, Summer School*, D. Touretzky, G. Hinton and T. Sejnowski,(Eds.), pp. 120-126, Carnegie Mellon University, Morgan Kaufman Publishers, Inc., 1989.

59. Kukich, K., "Backpropagation Topologies for Sequence Generation," *Proceedings of 1988 IEEE International Conference on Neural Networks*, I-301.

60. Kung, S. Y., and Hwang, J. N., "An Algebraic Projection Analysis for Optimal Hidden Units Size and Learning Rates in Backpropagation Learning," *Proceedings of 1988 IEEE International Conference on Neural Networks*, I-363.

61. Lang, K. J., and Witbrock, M. J., "Learning to Tell Two Spirals Apart," *Proceedings of the 1988 Connectionist Models, Summer School*, D. Touretzky, G. Hinton and T. Sejnowski (Eds.), pp. 52-61, Carnegie Mellon University, Morgan Kaufman Publishers, Inc., 1989.

62. Le Cun, Y., "A Theoretical Framework For Backpropagation," *Proceedings of the 1988 Connectionist Models, Summer School*, D. Touretzky, G. Hinton and T. Sejnowski (Eds.), pp. 21-28, Carnegie Mellon University, Morgan Kaufman Publishers, Inc., 1989.

63. Le Cun, Y., Galland, C. C., and Hinton, G. E., "GEMINI: Gradient Estimation Through Matrix Inversion After Noise Injection," in *Advances in Neural Information Processing Systems I*, D. S. Touretzky, Editor, Morgan Kaufmann Publishers, Inc., pp. 141, 1989.

64. Lendaris, G. G., "A Proposal for Indicating Quality of Generalization when Evaluating ANNS," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17-21, 1990, I-709.

65. Li, S.-D., "An Optimized Backpropagation with Minimum Norm Weights," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17-21, 1990, I-697.

66. Miikkulainen, R., and Dyer, M. G., "Forming Global Representations with Extended Backpropagation," *Proceedings of 1988 IEEE International Conference on Neural Networks*, I-285.

67. Minai, A. A., and Williams, R. D., "Acceleration of Backpropagation Through Learning Rate and Momentum Adaptation," *Proceedings of the International Joint Conference on Neural Networks*, Washington, D. C., Jan. 15-19, 1990, I-676.

68. Minai, A. A., and Williams, R. D., "Backpropagation Heuristics: A Study of the Extended Delta-Bar-Delta Algorithm," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17-21, 1990, I-595.

69. Moody, J., "Fast Learning in Multi-Resolution Hierarchies," *Advances in Neural Information Processing Systems I*, D. S. Touretzky, Editor, Morgan Kaufmann Publishers, Inc., 1989, 29.

70. Movellan, J. R., "Error Functions to Improve Noise Resistance and Generalization in Backpropagation Networks," *Proceedings of the International Joint Conference on Neural Networks*, Washington, D. C., 1990, I-557.

71. Namatame, A., "Backpropagation Learning with High-Order Functional Networks and Analyses of Its Internal Representation," *Proceedings of the International Joint Conference on Neural Networks*, Washington, D. C., Jan. 15-19, 1990, I-680.

72. Nevard, J. A., "Tree Net: A Dynamically Configureable Neural Net," *Proceedings of the International Joint Conference on Neural Networks*, Washington, D. C., Jan. 15-19, 1990, I-549.

73. Nguyen, D., and Widrow, B., "Improving the Learning Speed of 2-layer Neural Networks by Choosing Initial Values of the Adaptive Weights," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17-21, 1990, III-21.

74. Oden, G. C., "FuzzyProp: A Symbolic Superstrate for Connectionist Models," *Proceedings of 1988 IEEE International Conference on Neural Networks*, I-293.

75. Otwell, K., "Incremental Backpropagation Learning from Novelty-based Orthogonalization," *Proceedings of the International Joint Conference on Neural Networks*, Washington, D. C., Jan. 15-19, 1990, I-561.

76. Parker, D. P., "Optimal Algorithms For Adaptive Networks: Second Order Back Propagation, Second Order Direct Propagation, And Second Order Hebbian Learning," *Proceedings of the First IEEE International Conference on Neural Networks*, 1987, II-593.

77. Pemberton, J. C., and Vidal, J. J., "When Is the Generalized Delta Rule A Learning Rule," *Proceedings of 1988 IEEE International Conference on Neural Networks*, I-309.

78. Pineda, F. J., "Generalization of Backpropagation to Recurrent and Higher Order Neural Network," in *Neural Information Processing Systems*, D. Z. Anderson, Editor, American Institute of Physics, New York, pp. 602-611, 1988.

79. Radons, G., Schustrem H. G., and Wener, D., "Drift and Diffusion in Backpropagation Networks," in *Parallel Processing in Neural Systems and Computers*, R. Echmiller, G. hartmann, G. Hauske, Editors North-Holland, pp. 261, 1990.

80. Rezgui, A., and Tepedelenlioglu, N., "The Effect of the Slope of the Activation Function on the Backpropagation Algorithm," *Proceedings of the International Joint Conference on Neural Networks*, Washington, D. C., Jan. 15-19, 1990, I-707.

81. Robinson, A. J., and Fallside, F., "Static and Dynamic Error Propagation Networks with Application to Speech Coding," in *Neural Information Processing Systems*, D. Z. Anderson, Editor, American Institute of Physics, New York, 1988, 632-641.

82. Rohwer, R., and Forrest, B., "Training Time Dependence in Neural Networks," *Proceedings of the First IEEE International Conference on Neural Networks*, San Diego, 1987, II-701.

83. Samad, T. "Backpropagation Improvements Based on Heuristic Arguments," *Proceedings of the International Joint Conference on Neural Networks*, Washington, D. C., 1990, I-565.

84. Sandon, P. A., and Uhr, L. M., "A Local Interaction Heuristic for Adaptive Networks," *Proceedings of 1988 IEEE International Conference on Neural Networks*, I-317.

85. Sbarbaro, D., and Gawthrop, P. J., "Learning Complex Mappings By Stochastic Approximation," *Proceedings of the International Joint Conference on Neural Networks*, Washington, D. C., January, 1990, I-569.

86. Scheff, K., and Willey, J., "Simulation Results of Multiple Layer Difference Learning," *Proceedings of the First IEEE International Conference on Neural Networks*, San Diego, 1987, II-679.

87. Schiffmann, W., and Mecklengurg, K., "Genetic Generation of Backpropagation Trained Neural Networks," in *Parallel Processing in Neural Systems and Computers*, R. Echmiller, G. Hartmann, G. Hauske, Editors, North-Holland, pp. 205, 1990.

88. Schreibman, D. V., "Speeding Up Backpropagation By Gradient Correlation," *Proceedings of the International Joint Conference on Neural Networks*, Washington, D. C., 1990, I-723.

89. Shah, S., and Palmieri, F., "MEKA: A Fast, Local Algorithm for Training Feedforward Neural Networks," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17-21, 1990, III-41.

90. Shepanski, J. F., "Fast Learning in Artificial Neural Systems: Multilayer Perceptron Training Using Optimal Estimation," *Proceedings of 1988 IEEE International Conference on Neural Networks*, I-465.

91. Shimohara, K., Uchiyama, T., and Tokunaga, Y., "Backpropagation Networks for Event Driven Temporal Sequence Processing," *Proceedings of the IEEE International Conference on Neural Networks*, 1988, I-673.

92. Shmad, S., and Tesauro, G., "Scaling and Generalization in Neural Networks: A Case Study," *Proceedings of the 1988 Connectionist Models, Summer School*, D. Touretzky, G. Hinton and T. Sejnowski (Eds.), pp. 3-10, Carnegie Mellon University, Morgan Kaufman Publishers, Inc., 1989.

93. Shoemaker, P. , Carlin, M. J., and Shimabukuro, R. L., "Backpropagation Learning with Coarse Quantization of Weight Updates," *Proceedings of the International Joint Conference on Neural Networks*, Washington, D. C., January 1990, I-573.

94. Sietsma, J., and Dow, R. J. F., "Neural Net Pruning - Why and How," *Proceedings of 1988 IEEE International Conference on Neural Networks*, I-325.

95. Singhal, S., and Wu, Lance, "Training Multilayer Perceptrons with the Extended Kalman Algorithm," in *Advances in Neural Information Processing Systems I*, D. S. Touretzky, Editor, Morgan Kaufmann Publishers, Inc., 1989, 133.

96. Song, J., and Hassoun, M. H., "Learning with Hidden Targets," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17–21, 1990, III–93.

97. Soulie, F. F., Gallinari, P., Le Cun, Y., and Thiria, S., "Evaluation of Network Architectures on Test Learning Tasks," *Proceedings of the First IEEE International Conference on Neural Networks*, San Diego, 1987, II–653.

98. Stinchcombe, M., and White, H., "Approximating and Learning Unknown Mappings Using Multilayer Feedforward Networks with Bounded Weights," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17–21, 1990, III–7.

99. Stornetta, W. Scott, and Huberman, B. A., "An Improved Three-layer Back Propagation Algorithm," *Proceedings of the First IEEE International Conference on Neural Networks*, San Diego, 1987, II–637.

100. Sun, J., Grosky, W. I., and Hassoun, M. H., "A Fast Algorithm for Finding Global Minima of Error Functions in Layered Neural Networks," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17–21, 1990, I–715.

101. Tanaka, T., Naka, M., and Yoshida, K., "Improved Backpropagation Combined with LVQ," *Proceedings of the International Joint Conference on Neural Networks*, Washington, D. C., 1990, I–731.

102. Ultsch, A., Hannuschka, R., Hartmann, U., and Weber, V., "Learning of Control Knowledge for Symbolic Proofs with Backpropagation Networks," in *Parallel Processing in Neural Systems and Computers*, R. Echmiller, G. Hartmann, G. Hauske, Editors, North-Holland, pp. 499, 1990.

103. Von Lehmen, A., Paek, E. G., Liao, P. F., Marrakchi, A., and Patel, J. S., "Factors Influencing Learning by Backpropagation," *Proceedings of 1988 IEEE International Conference on Neural Networks*, I–335.

104. Wah, B. W., and Kriplani, H., "Resource Constrained Design of Artificial Neural Networks," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17–21, 1990, III–269.

105. Wan, E. A., "Temporal Backpropagation for FIR Neural Networks," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17–21, 1990, I–575.

106. Watrous, R. L., "Learning Algorithm For Connectionist Networks: Applied Gradient Methods Of Nonlinear Optimization," *Proceedings of the IEEE International Conference on Neural Networks*, 1987, II–619.

107. Werbos, P. J., "Backpropagation: Past and Future," *Proceedings of 1988 IEEE International Conference on Neural Networks*, I–343.

108. White, H., "Some Asymptotic Results for Backpropagation," *Proceedings of the IEEE First International Conference on Neural Networks*, 1987, III–261.

109. White, R. H., "The Learning Rate in Backpropagation Systems: An Application of Newton's Method," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17–21, 1990, I–679.

110. Wilensky, G. D., and Neuhaus, J. A., "Scaling of Backpropagation Training Time to Large Dimensions," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17-21, 1990, III-239.

111. Windrow, B., Winter, R. G., and Baxter, R. A., "Learning Phenomena In Layered Neural Networks," *Proceedings of the IEEE International Conference on Neural Networks*, 1987, II-411.

112. Woods, D., "Back and Counter Propagation Aberrations," *Proceedings of 1988 IEEE International Conference on Neural Networks*, I-473.

113. Xue, Q., Hu, Y., and Tompkins, W. J., "Analyses of the Hidden Units of Backpropagation Model by Singular Value Decomposition," *Proceedings of the International Joint Conference on Neural Networks*, Washington, D. C., 1990, I-739.

114. Yang, H., and Guest, C.C., "Linear Discriminants, Logic Functions, Backpropagation, and Improved Convergence," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17-21, 1990, III-287.

115. Yau, H. C., and Manry, M. T., "Sigma-Pi Implementation of a Nearest Neighbor Classifier," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17-21, 1990, III-825.

116. Ying, X., "Role of Activation Function on Hidden Units for Sample Recording in Three-Layer Neural Networks," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17-21, 1990, III-69.

117. Yoshimoto, S., "A Study on Artificial Neural Network Generalization Capability," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17-21, 1990, III-689.

118. Yu, Y. H., and Simmons, R. F., "Extra Output Biased Learning," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17-21, 1990, III-161.

119. Yu, Y. H., and Simmons, R. F., "Descending Epsilon in Backpropagation: A Technique for Better Generalization," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17-21, 1990, III-167.

2.5 Theoretical Analysis of Supervised Learning Models

The previously described approaches to the architectural determination of, and learning performance improvement on, the supervised learning models can be regarded as engineering approaches. On the other hand, as a learning paradigm, neural networks have their intrinsic properties, such as their dynamics, modeling capability, internal feature representation, and generalization characteristics. The purpose of these kinds of theoretical studies is to understand the underlying properties and rules that govern the behavior, operation and reasoning of different neural network learning models such that their application to real world problems would be well guided. A good example is the rigorous mathematical analysis of Rosenblatt's Perceptron by Minsky and Papert (1969) to expose the exact limitations of a class of computing machines that could seriously be considered as models of the brain. Mathematical analysis, though it has its limitations, in most of the cases, shows the elegance of logic as well as sober and rational thinking.

In a general sense, from a numerical analysis perspective, there are two salient features that govern the operation of neural networks: the stability associated with feedback recall and the convergence with supervised learning models. Global stability refers to the stabilization of the activation patterns of a network from any input pattern, and convergence refers to the ability to reduce error measurement of a system in a large enough time. Convergence can be represented in an absolute sense and in the mean square sense (Papoulis, 1965). All the theorems concerning neural network learning are based on the definition of global stability proposed by Lyapunov, which states that for all possible system inputs X to a dynamic system, if X is zero only at the origin, having the first derivative defined in a given domain and being upper-bounded, then the system that is defined by a Lyapunov energy function of the variables of X that maps n dimensions to one will converge and become globally stable for all the inputs X (Chetayav, 1961). There are three stability theorems for nonadaptive autoassociator (Cohen and Grossberg, 1983), adaptive autoassociator, and adaptive heteroassociator (Kosko, 1988) respectively (Simpson, 1990).

For multilayer feedforward networks, their general mapping ability has been proved by several researchers and has generated more enthusiasm and confidence since then. On the functional modeling, Hecht-Nielsen (1987) uses Kolmogorov's superposition theorem to generally support the modeling capability of a multilayer feedforward network; Gallant and White (1988) shows that a three layer network, with one hidden layer, is capable of embedding a Fourier analyzer by using the monotone cosine squasher; and recently, Hornik, et al. (1989) proved that multilayer feedforward networks are universal approximators. Issues like the complexity of loading shallow neural networks, estimation of neurons in the hidden layer, scaling property, and the mathematical theory of generalization, have also been proposed and extensively studied (Judd, 1990; Wolpert, 1990). Due to time limitation, the details of different approaches to the theoretical analysis, and that of different new learning theories are not summarized here. The reference listing following will provide a fairly good picture of the current state in this area.

References

1. Abu-Mostafa, Y. S., "Lower Bound for Connectivity in Local-Learning Neural Networks," *J. of Complexity 4*, pp. 246–255, 1988.

2. Ahmad, S., and Tesouro, G., "Scaling and Generalization in Neural Networks: A Case Study," in *Advances in Neural Information Processing Systems I*, D. S. Touretzky, Editor, Morgan Kaufmann Publishers, Inc., pp. 160, 1989.

3. Akingbehin, K., "A Decentralized Algorithm for Learning in Adaptive Networks," *Proceedings of 1988 IEEE International Conference on Neural Networks*, I–409.

4. Aleksander, I., "An Overview of Weightless Neural Nets," *Proceedings of the International Joint Conference on Neural Networks*, Washington, D. C., 1990, II–499.

5. Amari, S.-I., "On Mathematical Methods in the Theory of Neural Networks," *Proceedings of the First IEEE International Joint Conference on Neural Networks*, San Diego, 1987, III–3.

6. Anderson, C. H., and Abrahams, E., "The Bayes Connection," *Proceedings of the First IEEE International Joint Conference on Neural Networks*, San Diego, 1987, III–105.

7. Anlauf, J. K., and Biehl, M., "Properties of an Adaptive Perceptron Algorithm," in *Parallel Processing in Neural Systems and Computers*, R. Echmiller, G. Hartmann, G. Hauske, Editors, North-Holland, pp. 153, 1990.

8. Bastida, M. R., and Nazuno, J. F., "Brownian Motion Theory as A Physical Foundation for Feedback Neural Networks," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17-21, 1990, III-573.

9. Baum, E. B., "On the Capability of Multilayer Perceptrons," J. of Complexity 4, pp. 193–215, 1988.

10. Baum, E. B., and Haussler, D., "What Size Net Gives Valid Generalization?" *Advances in Neural Information Processing Systems I*, D. S. Touretzky, Editor, Morgan Kaufmann Publishers, Inc., pp. 81, 1989.

11. Belew, R. K., "Designing Appropriate Learning Rules for Connectionist Systems," *Proceedings of the First IEEE International Joint Conference on Neural Networks*, San Diego, 1987, II–479.

12. Belew, R. K., "Evolution, Leaning, and Culture: Computational Metaphors for Adaptive Algorithms," *Complex Systems 4*, pp. 11–49, 1990.

13. Benim, M., and Samuelides, M.,"Dynamical Properties of Neural Nets Using Competitive Activation Mechanisms," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17-21, 1990, III–541.

14. Bouju, A., Bourret, P., and Gaspin, C., "A Global State Driven Cooling Schema for Simulated Annealing Used with Boltzmann Machine," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17-21, 1990, III–577.

15. Brady, M., Raghavan, R., and Slawny, J., "Gradient Descent Fails to Separate," *Proceedings of 1988 IEEE International Conference on Neural Networks*, I–649.

16. Bruck, J., and Goodman, J. W., "On the Power of Neural Networks for Solving Hard Problems," in *Neural Information Processing Systems*, D. Z. Anderson, Editor, American Institute of Physics, New York, pp. 137–143, 1988.

17. Carley, L. R., "Presynaptic Neural Information Processing," in *Neural Information Processing Systems*, D. Z. Anderson, Editor, American Institute of Physics, New York, pp. 154–163, 1988.

18. Chetayav, N., *The Stability of Motion*, Pergamon Press, New York, 1961.

19. Cheung, K. F., Oh, S., Marks II, R. J., and Atlas, L. E., "Neural Net Associative Memories Based on Convex Set Projections," *Proceedings of the First IEEE International Conference on Neural Networks*, San Diego, 1987, III–245.

20. Chiueh, T.-D., and Goodman, R., "A Neural Network Classifier Based on Coding Theory," in *Neural Information Processing Systems*, D. Z. Anderson, Editor, American Institute of Physics, New York, pp. 174–183, 1988.

21. Chou, P. A., "The Capacity of the Kanerva Associative Memory is Exponential," in *Neural Information Processing Systems*, D. Z. Anderson, Editor, American Institute of Physics, New York, pp. 184–191, 1988.

22. Chover, J., "Phase Transitions in Neural Networks," in *Neural Information Processing Systems*, D. Z. Anderson, Editor, American Institute of Physics, New York, pp. 192–200, 1988.

23. Cohen, M., and Grossberg, S., "Absolute Stability of Global Pattern Formation and Parallel Memory Storage by Competitive Neural Networks," *IEEE Transactions on Systems, Man, and Cybernetics*, SMC–13, pp. 815–825, 1983.

24. Collins, J. W., "A Neural Network Based on Co–occurrence Probabilities," *Proceedings of the First IEEE International Joint Conference on Neural Networks*, San Diego, 1987, II–487.

25. Cotter, N. E., and Conwell, P. R., "Fixed Weight Networks Can Learn," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17-21, 1990, III–553.

26. Dahl, E. D., "Accelerated Learning Using the Generalized Delta Rule," *Proceedings of the First IEEE International Joint Conference on Neural Networks*, San Diego, 1987, II–523.

27. Dayhoff, J. E., "Regularity Properties in Pulse Transmission Networks," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17-21, 1990, III–621.

28. Denker, J. S., Schwartz, D., Wittner, B., Solla, S., Howard, R., Jackel, L., and Hopfield, J., "Large Automatic Learning, Rule Extraction, and Generalization," *Complex Systems 1*, pp. 877–922, 1987.

29. Denker, J. S., and Wittner, B.S., "Network Generality, Training Required, and Precision Required," in *Neural Information Processing Systems*, D. Z. Anderson, Editor, American Institute of Physics, New York, pp. 219–222, 1988.

30. Dreyfus, G., Guyon, I., Nadal, J. P., and Personnaz, L., "High Order Neural Networks for Efficient Associative Memory Design," in *Neural Information Processing Systems*, D. Z. Anderson, Editor, American Institute of Physics, New York, pp. 242–249, 1988.

31. Engel, J., "Teaching Feedforward Neural Networks by Simulated Annealing," *Complex Systems 2*, pp. 641–648, 1988.

32. Eppler, W., "Implementation of Fuzzy Production Systems with Neural Networks," in *Parallel Processing in Neural Systems and Computers*, R. Echmiller, G. Hartmann, G. Hauske, Editors, North-Holland, pp. 249, 1990.

33. Ersu, E., and Tolle, H., "Hierarchical Learning Control – An Approach with Neuron-Like Associative Memories," in *Neural Information Processing Systems*, D. Z. Anderson, Editor, American Institute of Physics, New York, pp. 249–261, 1988.

34. Fetz, E. E., "Correlational Strength and Computational Algebra of Synaptic Connections Between Neurons," in *Neural Information Processing Systems*, D. Z. Anderson, Editor, American Institute of Physics, New York, pp. 270–277, 1988.

35. Flesher, M., "The Hopfield Model with Multi-Level Neurons," in *Neural Information Processing Systems*, D. Z. Anderson, Editor, American Institute of Physics, New York, pp. 278–289, 1988.

36. Gaines, B. R., "Uncertainty as a Foundation of Computational Power in Neural Networks," *Proceedings of the First IEEE International Joint Conference on Neural Networks*, San Diego, 1987, III–51.

37. Gallant, A. R., and White, H., "There Exists a Neural Network that Does Not Make Avoidable Mistakes," *Proceedings of 1988 IEEE International Conference on Neural Networks*, I-657.

38. Gallinari, P., Thiria, S., and Soulie, F. F., "Multilayer Perceptrons and Data Analysis," *Proceedings of 1988 IEEE International Conference on Neural Networks*, I-391.

39. Gaudiano, P., "Temporal Patterns of Activity in Neural Networks," in *Neural Information Processing Systems*, D. Z. Anderson, Editor, American Institute of Physics, New York, pp. 297-300, 1988.

40. Gelenbe, E., and Stafylopatis, A., "Temporal Behavior of Neural Networks," *Proceedings of the First IEEE International Conference on Neural Networks*, San Diego, 1987, III-267.

41. Giles, C. L., Griffin, R. D., and Maxwell, T., "Encoding Geometric Invariances in Higher-Order Neural Networks," in *Neural Information Processing Systems*, D. Z. Anderson, Editor, American Institute of Physics, New York, pp. 301-310, 1988.

42. Goel, A., Ramanujam, J., and Sadayappan, P., "Towards a Neural Architecture for Abductive Reasoning," *Proceedings of 1988 IEEE International Conference on Neural Networks*, I-681.

43. Golden, R. M., "Probabilistic Characterization of Neural Model Computations," in *Neural Information Processing Systems*, D. Z. Anderson, Editor, American Institute of Physics, New York, pp. 310-316, 1988.

44. Gorse, D., Taylor, J. G., "Training Strategies for Probabilistic RAMs," in *Parallel Processing in Neural Systems and Computers*, R. Echmiller, G. Hartmann, G. Hauske, Editors, North-Holland, pp. 157, 1990.

45. Grossberg, S., and Gutowski, W., "Neural Dynamics of Decision Making Under Risk: Affective Balance Theory," *Proceedings of the First IEEE International Joint Conference on Neural Networks*, San Diego, 1987, III-31.

46. Grossman, T., Meir, R., and Domany, E., "Learning by Choice of Internal Representations," *Complex Systems 2*, pp. 555-575, 1988.

47. Haken, A., and Luby, M., "Steepest Descent Can Take Exponential time for symmetric Connection Networks," *Complex Systems 2*, pp. 191-196, 1988.

48. Hartley, R., and Szu, H., "A Comparison of the Computational Power of Neural Network Models," *Proceedings of the First IEEE International Joint Conference on Neural Networks*, San Diego, 1987, III-15.

49. Hecht-Nielsen, R., "Kolmogorov's Mapping Neural Network Existence Theorem," *Proceedings of the First IEEE International Joint Conference on Neural Networks*, San Diego, 1987, III-11.

50. Hennessey, K., and Farley, E., "The Gamma Network as an Interconnection Architecture for Neural Networks," *Proceedings of the First IEEE International Joint Conference on Neural Networks*, San Diego, 1987, III-131.

51. Hinton, G. E., and Nowlan, S. J., "How Learning Can Guide Evolution," *Complex Systems 1*, pp. 495-502, 1987.

52. Hinton, G. E., and McClelland, J. L., "Learning Representations by Recirculation," in *Neural Information Processing Systems*, D. Z. Anderson, Editor, American Institute of Physics, New York, pp. 358–366, 1988.

53. Huang, W. Y., and Lippmann, R. P., "Neural Net and Traditional Classifiers," in *Neural Information Processing Systems*, D. Z. Anderson, Editor, American Institute of Physics, New York, pp. 387–396, 1988.

54. Israel, P., and Koutsougeras, C., "An Annealing Approach to Associative Recall in the CBM Model," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17-21, 1990, III–633.

55. Jang, J.–S., Lee, S.–Y., and Sin, S.–Y., "An Optimization Network for Matrix Inversion," in *Neural Information Processing Systems*, D. Z. Anderson, Editor, American Institute of Physics, New York, pp. 397–401, 1988.

56. Josin, G. M., "Response Learning," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17-21, 1990, III–599.

57. Judd, S., "On the Complexity of Loading Shallow Neural Networks," *J. Of Complexity 4*, pp. 177–192, 1988.

58. Judd, S., Neural Network Design and the Complexity of Learning, The MIT Press, Cambridge, MA, 1990.

59. Karayiannis, N. B., Venetsanopoulos, A. N., "Efficient Learning Algorithms for Single Layered Neural Networks," in *Parallel Processing in Neural Systems and Computers*, R. Echmiller, G. Hartmann, G. Hauske, Editors, North-Holland, pp. 169, 1990.

60. Keeler, J. D., "Comparison of Information Capacity of Hebbian Neural Networks," *Proceedings of the First IEEE International Conference on Neural Networks*, San Diego, 1987, III–253.

61. Kimura, T. D., "An Algebraic Proof for Backpropagation in Acyclic Neural Networks," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17-21, 1990, III–547.

62. Klassen, M., Pao, Y. H., and Chen, V., "Characteristics of the Functional Link Net: A Higher Order Delta Rule Net," *Proceedings of 1988 IEEE International Conference on Neural Networks*, I–507.

63. Kohonen, T., "Internal Representations and Associative Memory," in *Parallel Processing in Neural Systems and Computers*, R. Echmiller, G. Hartmann, G. Hauske, Editors, North-Holland, pp. 177, 1990.

64. Kosko, B., "Feedback Stability and Unsupervised Learning," *Proceedings of the 1988 IEEE International Conference on Neural Networks*, San Diego, 1988, I–141.

65. Koutsougeras, C., and Papachristou, C. A., "Training of a Neural Network for Pattern Classification Based on Entropy Measure," *Proceedings of 1988 IEEE International Conference on Neural Networks*, I–247.

66. Kuh, A., "Performance Measures for Associative Memories that Learn and Forget," in *Neural Information Processing Systems*, D. Z. Anderson, Editor, American Institute of Physics, New York, pp. 432–441, 1988.

67. Kurten, K. E., "Dynamical Learning in Networks with Sparse Connectivity," in *Parallel Processing in Neural Systems and Computers*, R. Echmiller, G. Hartmann, G. Hauske, Editors, North-Holland, pp. 191, 1990.

68. Laird, P. D., "A Survey of Computational Learning Theory," in *Formal Techniques in Artificial Intelligence – A Sourcebook*, R. B. Banerji, Editor, Elsevier Science Publishers B. V., pp. 173, 1990.

69. Lapedes, A., and Farber, R., "How Neural Nets Work", in *Neural Information Processing Systems*, D. Z. Anderson, Editor, American Institute of Physics, New York, pp. 442–456, 1988.

70. Lee, S., and Kil, R. M., "Multilayer Feedforward Potential Function Network," *Proceedings of 1988 IEEE International Conference on Neural Networks*, I–161.

71. Leong, H. M., "Optimization with Artificial Neural Network Systems: A Mapping Principle and a Comparison to Gradient Based Methods," in *Neural Information Processing Systems*, D. Z. Anderson, Editor, American Institute of Physics, New York, pp. 474–484, 1988.

72. Leshno, M. "On the Computational Power of Neural Networks and Neural Automata," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17-21, 1990, III–535.

73. Linsker, R., "Towards an Organizing Principle for a Layered Perceptual Network," in *Neural Information Processing Systems*, D. Z. Anderson, Editor, American Institute of Physics, New York, pp. 485–494, 1988.

74. Linsker, R., "An Application of the Principle of Maximum Information Preservation to Linear System," in *Advances in Neural Information Processing Systems I*, D. S. Touretzky, Editor, Morgan Kaufmann Publishers, Inc., pp. 195, 1989.

75. Littlestone, N., "Learning in a Layered Network with Many Fixed-function Hidden Nodes," *Proceedings of the First IEEE International Joint Conference on Neural Networks*, San Diego, 1987, II–471.

76. Loos, H. G., "Reflexive Associative Memories," in *Neural Information Processing Systems*, D. Z. Anderson, Editor, American Institute of Physics, New York, pp. 495–504, 1988.

77. MacDonald, B., "Connecting to the Past," in *Neural Information Processing Systems*, D. Z. Anderson, Editor, American Institute of Physics, New York, pp. 505–514, 1988.

78. Marks II, R. J., Atlas, L. E., Oh, S., and Ritcey, J. A., "The Performance of Convex Set Projection Based Neural Networks," in *Neural Information Processing Systems*, D. Z. Anderson, Editor, American Institute of Physics, New York, pp. 534–543, 1988.

79. Matsuo, K., "Self-Organizations of Cooperative and Competitive Networks in Ecological Paradigm," *Proceedings of the First IEEE International Conference on Neural Networks*, San Diego, 1987, III–277.

80. Mazaika, P. K., "A Mathematical Model of the Boltzmann Machine," *Proceedings of the First IEEE International Joint Conference on Neural Networks*, San Diego, 1987, III–157.

81. McCuloch, N., "Recurrent Networks for Learning Stochastic Sequences," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17-21, 1990, III–561.

82. Michel, A. N., Farrell, J. A., and Porod, W., "Stability Results for Neural Networks," in *Neural Information Processing Systems*, D. Z. Anderson, Editor, American Institute of Physics, New York, pp. 554–565, 1988.

83. Milligan, D. K., and Wilson, M. J. D., "Requirements for the Synthesis of Boolean Neural Networks," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17-21, 1990, III–567.

84. Minsky, M., and Papert, S., *Perceptrons*, The MIT Press, Cambridge, MA, 1969; The second Edition, 1989.

85. Mozer, M. C., and Smolensky, P., "Skeletonization: A Technique for Trimming the Fat From A Network via Relevance Assessment," *Advances in Neural Information Processing Systems I*, D. S. Touretzky, Editor, Morgan Kaufmann Publishers, Inc., pp. 107, 1989.

86. Murray, A. F., Smith, A. V. W., and Butler, Z. F., "Bit-Serial Neural Networks," in *Neural Information Processing Systems*, D. Z. Anderson, Editor, American Institute of Physics, New York, pp. 573–583, 1988.

87. Nguyen, D., and Holt, F., "Stochastic Processing in A Neural Network Application," *Proceedings of the First IEEE International Conference on Neural Networks*, San Diego, 1987, III–281.

88. Nguyen, D. D., and Lee, J. S. J., "A New LMS–Based Algorithm for Rapid Adaptive Classification in Dynamic Environments: Theory and Preliminary Results," *Proceedings of 1988 IEEE International Conference on Neural Networks*, I–455.

89. Niznik, C. A., Hoss, W., and Watts, C., "Computer Networking Topological and Parametric Representations for Functional Neural Networks," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17-21, 1990, III–583.

90. Noest, A. J., "Phasor Neural Networks," in *Neural Information Processing Systems*, D. Z. Anderson, Editor, American Institute of Physics, New York, pp. 584–591, 1988.

91. Oja, E., and Kohonen, T., "The Subspace Learning Algorithm as a Formalism for Pattern Recognition and Neural Networks," *Proceedings of 1988 IEEE International Conference on Neural Networks*, I–277.

92. Omohundro, S. M., "Efficient Algorithms with Neural Network Behavior," *Complex Systems 1*, pp. 273–347, 1987.

93. Ossen, A., "Top-Down Learning in Modular Feed Forward Networks," in *Parallel Processing in Neural Systems and Computers*, R. Echmiller, G. Hartmann, G. Hauske, Editors, North-Holland, pp. 201, 1990.

94. Parberry, I., "A Primer on the Complexity Theory of Neural Networks," in *Formal Techniques in Artificial Intelligence – A Sourcebook*, R. B. Banerji, Editor, Elsevier Science Publishers B. V., pp. 217, 1990.

95. Papoulis, A., *Probability, Random Variables, and Stochastic Processes*, McGraw-Hill, New York, 1965.

96. Parker, D. B., Gluck, M., and Reifsnider, E. S., "Learning with Temporal Derivatives in Pulse-Coded Neuronal Systems," in *Advances in Neural Information Processing Systems I*, D. S. Touretzky, Editor, Morgan Kaufmann Publishers, Inc., pp. 195, 1989.

97. Peterson, C., and Anderson, J. R., "A Mean Field Theory Learning Algorithm for Neural Networks," *Complex Systems* 1, pp. 995–1019, 1987.

98. Petsche, T., and Dickinson, B. W., "A Trellis-Structured Neural Network," in *Neural Information Processing Systems*, D. Z. Anderson, Editor, American Institute of Physics, New York, pp. 592–601, 1988.

99. Pineda, F. J., "Dynamics and Architecture for Neural Computation," *J. of Complexity 4*, pp. 216–245, 1988.

100. Platt, J. C., and Barr, A. H., "Constrained Differential Optimization," in *Neural Information Processing Systems*, D. Z. Anderson, Editor, American Institute of Physics, New York, pp. 612–621, 1988.

101. Priebe, C. E., and Sung, C. H., "Temporal Pattern Recognition," *Proceedings of 1988 IEEE International Conference on Neural Networks*, I–689.

102. Psaltis, D., and Neifeld, M., "The Emergence of Generalization in Networks with Constrained Representations," *Proceedings of 1988 IEEE International Conference on Neural Networks*, I–371.

103. Qian, S., Lee, Y. C., Jones, R. D., Barnes, C. W., and Lee, K., "Function Approximation with An Orthogonal Basis Net," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17-21, 1990, III–605.

104. Ryan, T. W., "The Resonance Correlation Network," *Proceedings of 1988 IEEE International Conference on Neural Networks*, I–673.

105. Saad, D., and Marom, E., "Learning by Choice of Internal Representations: An Energy Minimization Approach," *Complex Systems 4*, pp. 107–118, 1990.

106. Scofield, C. L., "Learning Internal Representations in the Coulomb Energy Network," *Proceedings of 1988 IEEE International Conference on Neural Networks*, I–271.

107. Shrier, S., Barron, R. L., and Gilstrap, L. O., "Polynomial and Neural Networks: Analogies and Engineering Applications," *Proceedings of the First IEEE International Joint Conference on Neural Networks*, San Diego, 1987, II–431.

108. Silverman, R. H., and Noetzel, A. S., "Time Sequential Self Organization of Hierarchical Neural Networks," in *Neural Information Processing Systems*, D. Z. Anderson, Editor, American Institute of Physics, New York, pp. 709–714, 1988.

109. Simpson, P. K., Artificial Neural Systems: Foundations, Paradigms, Applications and Implementations, Pergamon Press, New York, 1990.

110. Smolensky, P., "Analysis of Distributed Representation of Constituent Structure in Connectionist Systems," in *Neural Information Processing Systems*, D. Z. Anderson, Editor, American Institute of Physics, New York, pp. 730–739, 1988.

111. Somani, A. K., and Penla, N., "Compact Neural Network," *Proceedings of the First IEEE International Conference on Neural Networks*, San Diego, 1987, III–191.

112. Stafylopatis, A., Dikaiakos, M., Kontoravdis, D., "Spatial Organization of Neural Networks: A Probabilistic Modeling Approach," in *Neural Information Processing Systems*, D. Z. Anderson, Editor, American Institute of Physics, New York, pp. 740–749, 1988.

113. Stornetta, W. S., Hogg, T., and Huberman, B. A., "A Dynamical Approach to Temporal Pattern Processing," in *Neural Information Processing Systems*, D. Z. Anderson, Editor, American Institute of Physics, New York, pp. 750–759, 1988.

114. Sun, G. Z., Lee, Y. C., and Chen, H. H., "A Novel Net that Learns Sequential Decision Process," in *Neural Information Processing Systems*, D. Z. Anderson, Editor, American Institute of Physics, New York, pp. 760–766, 1988.

115. Sun, G. Z., Chen, H. H., and Lee, Y. C., "Parallel Sequential Induction Network: A New Paradigm of Neural Network Architecture," *Proceedings of 1988 IEEE International Conference on Neural Networks*, I–489.

116. Suzuji, H., and Arimoto, S., "Self-Organization of Associative Database and Its Applications," in *Neural Information Processing Systems*, D. Z. Anderson, Editor, American Institute of Physics, New York, pp. 767–774, 1988.

117. Tagliarini, G. A., and Page, E. W., "A Neural-Network Solution to the Concentrator Assignment Problem," in *Neural Information Processing Systems*, D. Z. Anderson, Editor, American Institute of Physics, New York, pp. 775–783, 1988.

118. Tawel R., "Does the Neuron Learn Like the Synapse," in *Advances in Neural Information Processing Systems I*, D. S. Touretzky, Editor, Morgan Kaufmann Publishers, Inc., pp. 169, 1989.

119. Tesauro, G., "Scaling Relationships in Backpropagation Learning: Dependence on Training Set Size," *Complex Systems 1*, pp. 367–372, 1987.

120. Tesauro, G., "Connectionist Learning of Expert Preferences by Comparison Training," *Advances in Neural Information Processing Systems I*, D. S. Touretzky, Editor, Morgan Kaufmann Publishers, Inc., pp. 99, 1989.

121. Van den Bout, D. E., and Miller, T. K., "A Stochastic Architecture for Neural Nets," *Proceedings of 1988 IEEE International Conference on Neural Networks*, I–481.

122. Vidal, J. J., and Haggerty, J., "Synchronization in Neural Nets," in *Neural Information Processing Systems*, D. Z. Anderson, Editor, American Institute of Physics, New York, pp. 824–829, 1988.

123. Vidyasagar, M., "An Analysis of the Flows of Neural Networks with Linear Interconnections," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17-21, 1990, III–523.

124. Widrow, B., Winter, R. G., and Baxter, R. A., "Learning Phenomena in Layered Neural Networks," *Proceedings of the First IEEE International Joint Conference on Neural Networks*, San Diego, 1987, II–411.

125. Wieland, A., and Leighton, R., "Geometric Analysis of Neural Network Capabilities," *Proceedings of the First IEEE International Conference on Neural Networks*, San Diego, 1987, III–385.

126. Windheuser, C., Kindermann, H., "Competitive Sequence Learning," in *Parallel Processing in Neural Systems and Computers*, R. Echmiller, G. Hartmann, G. Hauske, Editors, North-Holland, pp. 217, 1990.

127. Windecker, R. C., "Learning in Networks of Nondeterministic Adaptive Logic Elements," in *Neural Information Processing Systems*, D. Z. Anderson, Editor, American Institute of Physics, New York, pp. 840–849, 1988.

128. Winter, R., and Widrow, B., "Madaline Rule II: A Training Algorithm for Neural Networks," *Proceedings of 1988 IEEE International Conference on Neural Networks*, I–401.

129. Wittner, B. S., and Denker, J. S., "Strategies for Teaching Layered Networks Classification Tasks," in *Neural Information Processing Systems*, D. Z. Anderson, Editor, American Institute of Physics, New York, pp. 850–859, 1988.

130. Wittle, P., "The Exact Evaluation of Memory Capacity for New Stochastic Versions of the Hopfield Net," in *Parallel Processing in Neural Systems and Computers*, R. Echmiller, G. Hartmann, G. Hauske, Editors, North-Holland, pp. 269, 1990.

131. Wolpert, D. H., "A Mathematical Theory of Generalization," *Complex Systems 4*, pp. 151–249, 1990.

132. Yair, E., and Gersho, A., "The Boltzmann Perceptron Network: A Multi-Layered Feedforward Network Equivalent to the Boltzmann Machine," *Advances in Neural Information Processing Systems I*, D. S. Touretzky, Editor, Morgan Kaufmann Publishers, Inc., pp. 116, 1989.

2.6 Recurrent Networks

Recurrent networks have emerged from the need to internally model time factors, time varying behaviors, and sequential events. The learning model gets its name due to the very existence of recurrent or feedback links in the network. Recurrent networks usually assume a multilayer architecture or without clear distinction from unlayered form in the network with fully recurrent links. Sometimes, recurrent networks are referred to as recurrent backpropagation networks because of the close relation on the backpropagation learning algorithm (Pineda, 1987, 1988).

The representation of sequential events can be done with a time windowing scheme (Sejnowski and Rosenberg, 1986) or using a crude version of backpropagation through time (Rumelhart, et al., 1986) in which the recurrent network is unfolded into a multilayer feedforward network that grows by one layer on each time step. Backpropagation through time only works well when the time structure of the problem is known a priori. For the time windowing scheme, as has been pointed out by Elman (1988), following are some drawbacks: 1) Some interface mechanisms are needed to buffer the input, 2) The approach does not easily distinguish relative temporal position from an absolute temporal position, and 3) The shift register imposes a rigid limit on the duration of patterns and the length of the input vector is fixed. All in all, the time is represented explicitly and the sequentiality is enforced onto the network rather than internally constructed. To overcome those shortcomings, schemes are proposed such that the representation of time is accomplished by the effect that it has on processing. To date, there are basically three kinds of recurrent networks proposed: Jordan's Network (Jordan, 1986); Elman's Network (1988); and Fully Recurrent Networks (Williams and Zipser, 1989). The application of recurrent networks has covered an extensive domain including language processing (Behme, 1990; Giles, et al., 1990; Grajski, et al., 1990; Liu, et. al., 1990; Stolche, 1990), processing of time dependent parameters (Blumenfeld, 1990), pattern recognition and statistical classification (Wong and Vieth, 1990), learning stochastic sequence (McCuloch, 1990), vision (Qian and Sejnowski, 1989), and solving constraint problems (Schaller, 1990).

In the following paragraphs, the architecture and operation of the three basic recurrent networks are described and some references on the theoretical analysis of recurrent networks and their applications are also listed.

2.6.1 Jordan's Network

A Jordan network is a layered feedforward network with recurrent connections from the output layer to a section of the input layer. The recurrent connections copy the output at previous time to the input of the current time so that the hidden units see its own previous output and this knowledge then influences the subsequent behavior if only one hidden layer is used in the network. The recurrent connections are not trainable so that the recurrent network can be directly trained with the standard backpropagation learning algorithm. However, the presence of nontrainable recurrent connections limits the richness of time sequence representation. The architecture of a Jordan Network is shown in Fig . 2.5.

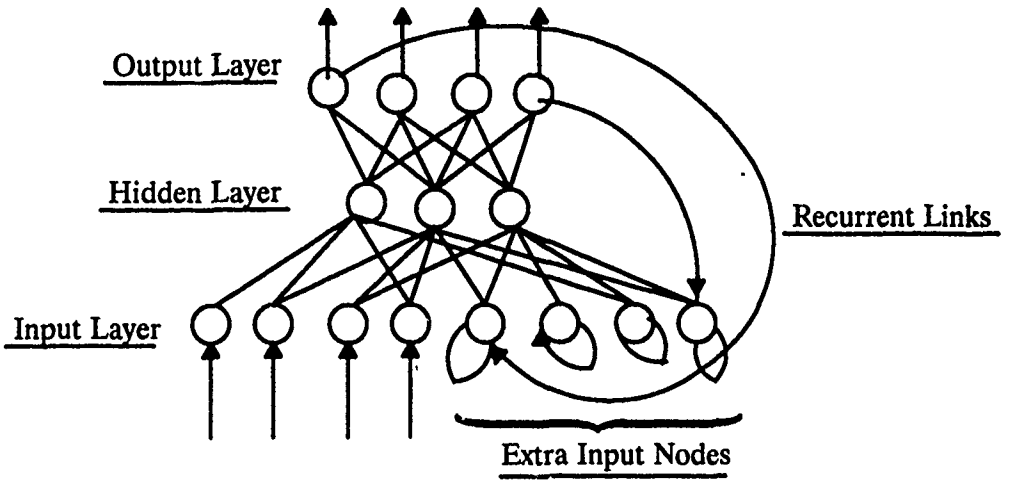


Figure 2.5 – Architecture of a Jordan Recurrent Network

2.6.2 Elman's Network

An Elman network is a modification of the Jordan network by introducing a set of context units in the input layer and making the recurrent connections from hidden layer to the context units in the input layer. The architecture of an Elman network is shown in Fig. 2.6. Like that in Jordan's network, the recurrent links are not trainable and the activations of context units at current time are merely a copy of those of the hidden units at previous time if a three layer feedforward network with one hidden layer is used. The training of the network is accomplished via backpropagation learning algorithm and the initial activations of the context units are set at 0.5 when the activation function is bounded in 0.0 to 1.0.

Though Eleman's modification on the Jordan network appears minor at first, it has infused a new representation scheme into the network. Because the features of the input-output are represented in the hidden units, the context units actually supply the network with a state identification for its previous state. The network still has the shortcoming of a Jordan network and the feature captured in the context units may not be crisp enough to give well defined state identification if the training data sets are noise contaminated.

2.6.3 Fully Recurrent Networks

In a fully recurrent network, the concept of layering is lost because each neuron is connected to every other neuron in the network and each one functions like both input and output units. For this kind of network, a gradient following learning procedure called real time recurrent learning (RTRL) has been proposed to suit the architectural complexity of the network (William and Zipser, 1989). With the use of RTRL, the network runs continually in the sense that they sample their inputs on every update cycle, and any unit can receive training signals on any cycle. In addition, it can solve problems

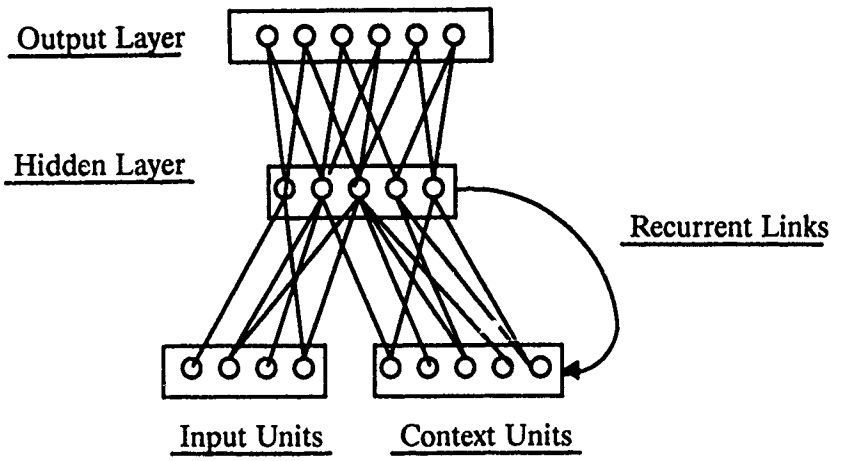


Figure 2.6 – Architecture of an Elman Recurrent Network

requiring an input of arbitrary length and it correctly does credit assignment to past events. The major drawbacks of using this kind of network via RTRL are that it is computationally very expensive to train the network and the learning procedure for RTRL is a non-local method.

2.6.4 Others' Work

There are, of course, many improvements and new schemes based on Jordan and Elman's work. Mozer (1988) made some improvement on the performance of the network by adding a layer of units that each gave a single self-recurrent connection that is trained by a true gradient-following learning rule. Pearlmutter (1988) proposed a scheme to improve the performance of backpropagation through time and Almeida (1987), Pineda (1988), as well as Rohwer & Forrest (1987) all derived various versions of the recurrent network in which the network's actual and desired dynamics settling to a fixed equilibrium state on each training cycle. Recently, a higher order recurrent network has been proposed by Giles, et al. (1990).

It can easily be seen that recurrent networks will have strong potential in modeling the time varying behaviors in engineering. In material modeling, recurrent networks have been proposed to model the mechanical behavior of engineering materials under cyclic loading (Wu, 1990). The challenge is on the identification of suitable problems and also the development of new and efficient learning algorithms for this kind of network. The reference listing provides information on both theoretical analysis and applications.

References

1. Almeida, L. B., "A Learning Rule for Asynchronous Perceptrons with Feedback in A Combinatorial Environment," in *Proceedings of the IEEE First International Conference on Neural Networks*, San Diego, 1987, II-609.

2. Bauer, H. U., and Geisel, T., "Dynamics of Signal Processing in Feedback Multilayer Perceptrons," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17-21, 1990, III-131.

3. Bauer, H. U., and Geisel, T., "Sequence Analysis in Feeedback Multilayer Perceptrons," in *Parallel Processing in Neural Systems and Computers*, R. Echmiller, G. Hartmann, G. Hauske, Editors, North-Holland, pp. 375, 1990.

4. Behme, H., "A Neural Net for Recognition and Storing of Spoken Words," in *Parallel Processing in Neural Systems and Computers*, R. Echmiller, G. Hartmann, G. Hauske, Editors, North-Holland, pp. 379, 1990.

5. Bell, T., "Sequential Processing Using Attractor Transitions," *Proceedings of the 1988 Connectionist Models, Summer School*, D. Touretzky, G. Hinton and T. Sejnowski (Eds.), pp. 93-102, Carnegie Mellon University, Morgan Kaufman Publishers, Inc., 1989.

6. Bereson, A. L., "Performance of Real-time Recurrent Learning Under Sub-optimal Conditions," *preprint*, Dept. of Electrical and Computer Engineering, University of Illinois, 1990.

7. Blumenfeld, B., "A Connectionist Approach to the Processing of Time Dependent Medical Parameters," *Proceedings of the International Joint Conference on Neural Networks*, Washington, D. C., January 1990, II-575.

8. Bourlard, H., and Wellekens, C. J., "Links between Markov Models and Multilayer Perceptrons," in *Advances in Neural Information Processing Systems I*, D. S. Touretzky, Editor, Morgan Kaufmann Publishers, Inc., pp. 503, 1989.

9. Brown, G. D. A., "Short-Term Memory Capacity Limitations in Recurrent Speech Production and Perception Networks," *Proceedings of the International Joint Conference on Neural Networks*, Washington, D. C., January 1990, I-43.

10. Elman, J. L., "Finding Structure in Time," *CRL Report 8801*, Center for Research in Language, University of California, San Diego, 1988.

11. Gately, M. T., "Cycles: A Simulation Tool for Studying Cyclic Neural Networks,"in *Neural Information Processing Systems*, D. Z. Anderson, Editor, American Institute of Physics, New York, pp. 290-296, 1988.

12. Gaudiano, P., and MacGregor, R., "Dynamic Activity and Memory Traces in Computer-Simulated Recurrently Connected Neural Networks," *Proceedings of the First IEEE International Conference on Neural Networks*, San Diego, 1987, II-177.

13. Gherrity, M., "A Learning Algorithm for Analog, Fully Recurrent Neural Networks," *Proceedings of the International Joint Conference on Neural Networks*, Washington, D. C., 1989, I-643.

14. Giles, C. L., Sun, G.Z., Chen, H. H., Lee, Y. C., and Chen, D., "Higher Order Recurrent Networks and Grammatical Inference," *Preprint*, Department of Physics and Astronomy and Institute for Advanced Computer Studies, University of Maryland, College Park, MD, 1990.

15. Goebel, R., "Learning Symbol Processing with Recurrent Networks," in *Parallel Processing in Neural Systems and Computers*, R. Eckmiller, G. Hartmann, G. Hauske, Editors, North-Holland, Elsevier Science Publishers B. V., pp. 157, 1990.

16. Grajski, K. A., Witmer, D. P., and Chen, C., "A Preliminary Note on Training Static and Recurrent Neural Networks for Word Level Speech Recognition," *Proceedings of the International Joint Conference on Neural Networks*, Washington, D. C., January 1990, II-245.

17. Hutton, L. V., Sigillitom V. G., and Egeth, H., "Experiments on Constructing A Cognitive Map: A Neural Network Model of a Robot that Daydreams," *Proceedings of the International Joint Conference on Neural Networks*, Washington, D. C., January 1990, I-223.

18. Jain, A. N., and Waibel, A. H., "Incremental Parsing by Modular Recurrent Connectionist Networks," in *Advances in Neural Information Processing Systems II*, D. S. Touretzky, Editor, Morgan Kaufmann Publishers, Inc., pp. 364, 1990.

19. Jordan, M. I., *The Learning of Representations for Sequential Performance*, Ph.D. Dissertation, University of California, San Diego, 1985.

20. Jordan, M. I., "Serial Order: A Parallel Distributed Processing Approach," *ICS Report 8604*, Institute for Cognitive Science, University of California, San Diego, 1986.

21. Jordan, M. I., "Supervised Learning and Systems with Excess Degrees of Freedom," *Proceedings of the 1988 Connectionist Models, Summer School*, D. Touretzky, G. Hinton and T. Sejnowski (Eds.), pp. 62–75, Carnegie Mellon University, Morgan Kaufman Publishers, Inc., 1989.

22. Kiwakewski, F., and Strube, H. W., "Word Recognition with A Recurrent Neural Network," in *Parallel Processing in Neural Systems and Computers*, R. Echmiller, G. Hartmann, G. Hauske, Editors, North-Holland, pp. 391, 1990.

23. Kuan, C.-M., "Learning Recurrent Networks by the Dynamic and Standard Backpropagation Methods," *Preprint*, Department of Economics, University of California, San Diego, 1989.

24. Liu, Y. D., Sun, G. Z., Chen, H. H., Lee, Y. C., and Giles, C. L., "Grammatical Inference and Neural Network State Machines," *Proceedings of the International Joint Conference on Neural Networks*, Washington, D. C., January 1990, I-285.

25. Ludermir, T. B., "A Feedback Network for Temporal Pattern Recognition," in *Parallel Processing in Neural Systems and Computers*, R. Echmiller, G. Hartmann, G. Hauske, Editors, North-Holland, pp. 395, 1990.

26. McCuloch, N., "Recurrent Networks for Learning Stochastic Sequences," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17-21, 1990.

27. Mjolsness, E., Sharp, D. H., and Alpert, B. K., "Recursively Generated Neural Networks," *Proceedings of the First IFEE International Conference on Neural Networks*, San Diego, 1987, III-165.

28. Mozer, M. C., "A Focused Backpropagation Algorithm for Temporal Pattern Recognition," *Technical Report*, University of Toronto, Departments of Psychology and Computer Science, 1988.

29. Norrod, F. E., O'Neill, M. D., and Gat, E., "Feedback-Induced Sequentiality in Neural Networks," in *Proceedings of the IEEE First International Conference on Neural Networks*, San Diego, 1987, II-251.

30. Peralmutter, B. A., "Learning State Space Trajectories in Recurrent Neural Networks," *Proceedings of the 1988 Connectionist Models, Summer School*, D. Touretzky, G. Hinton and T. Sejnowski (Eds.), pp. 113–119, Carnegie Mellon University, Morgan Kaufman Publishers, Inc., 1989.

31. Pineda, F. J., "Generalization of Backpropagation to Recurrent Neural Networks," *Phys. Rev. Lett*,. Vol. 18, pp. 2299, 1987.

32. Pineda, F. J., "Generalization of Backpropagation to Recurrent and Higher Order Neural networks," in *Neural Information Processing Systems*, D. Z. Anderson, Editor, American Institute of Physics, New York, pp. 602–611, 1988.

33. Pineda, F. J., "Time Dependent Adaptive Neural Networks," in *Advances in Neural Information Processing Systems II*, D. S. Touretzky, Editor, Morgan Kaufmann Publishers, Inc., pp. 710, 1990.

34. Pollack, J. B., "Implications of Recursive Distributed Representations," in *Advances in Neural Information Processing Systems I*, D. S. Touretzky, Editor, Morgan Kaufmann Publishers, Inc., pp. 527, 1989.

35. Qian, N., and Sejnowski, T. J., "Learning to Solve Random-Dot Stereograms of Dense and Transparent Surfaces with Recurrent Backpropagation," *Proceedings of the 1988 Connectionist Models, Summer School*, D. Touretzky, G. Hinton and T. Sejnowski (Eds.), pp. 435–443, Carnegie Mellon University, Morgan Kaufman Publishers, Inc., 1989.

36. Rohwer, R., and Forrest, B., "Training Time-Dependence in Neural Networks," in *Proceedings of the IEEE First International Conference on Neural Networks*, San Diego, 1987, II–701.

37. Rohwer, R., "The Moving Targets Training Algorithm," in *Advances in Neural Information Processing Systems II*, D. S. Touretzky, Editor, Morgan Kaufmann Publishers, Inc., pp. 558, 1990.

38. Rosen, B. E., Goodwin, J. M., and Vidal J. J., "Learning by State Recurrence Detection," in *Neural Information Processing Systems*, D. Z. Anderson, Editor, American Institute of Physics, New York, pp. 642–651, 1988.

39. Rumelhart, D., and McClelland, J. (Eds.), *Parallel Distributed Processing: Explorations in the Microstructure of Cognition; Vol. 1: Foundations*, The MIT Press, Cambridge, Mass., 1986.

40. Salam, F. M. A., and Bai, S., "A Feedback Neural Network with Supervised Learning," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17–21, 1990, III–263.

41. Schaller, H. N., "Solving Constraint Problems Using Feedback Neural Networks," in *Parallel Processing in Neural Systems and Computers*, R. Echmiller, G. Hartmann, G. Hauske, Editors, North-Holland, pp. 265, 1990.

42. Schmidhuber, J., "Temporal-difference-driven Learning in Recurrent Networks," in *Parallel Processing in Neural Systems and Computers*, R. Echmiller, G. Hartmann, G. Hauske, Editors, North-Holland, pp. 209, 1990.

43. Schmidhuber, J., "Recurrent Networks Adjusted by Adaptive Critics," *Proceedings of the International Joint Conference on Neural Networks*, Washington, D. C., January 1990, I–719.

44. Servan-Schreiber, D., Cleeremans, A., and McClelland, J. L., "Encoding Sequential Structure in Simple Recurrent Networks," in *Advances in Neural Information Processing Systems I*, D. S. Touretzky, Editor, Morgan Kaufmann Publishers, Inc., pp. 643, 1989.

45. Simard, P. Y., Ottaway, M. B., and Ballard, D. H., "Analysis of Recurrent Backpropagation," *Proceedings of the 1988 Connectionist Models, Summer School*, D. Touretzky, G. Hinton and T. Sejnowski (Eds.), pp. 103–112, Carnegie Mellon University, Morgan Kaufman Publishers, Inc., 1989.

46. Simard, P. Y., Ottaway, M. B., and Ballard, D. H., "Fixed Point Analysis for Recurrent Networks," in *Advances in Neural Information Processing Systems I*, D. S. Touretzky, Editor, Morgan Kaufmann Publishers, Inc., pp. 149, 1989.

47. Smith, A. W., and Zipser, D., "Encoding Sequential Structure: Experience with the Real Time Recurrent Learning Algorithm," *Proceedings of the International Joint Conference on Neural Networks*, Washington, D. C., 1989, I-645.

48. Stolcke, A., "Learning Feature-based Semantics with Simple Recurrent Networks," *TR–90–015*, International Computer Science Institute, Berkeley, California, April 1990.

49. Tsung, F.-S., and Cottrell, G. W., "A Sequential Adder Using Recurrent Networks," *Proceedings of the International Joint Conference on Neural Networks*, Washington, D. C., 1989, II-133.

50. Williams, R. J., and Zipser, D., "A Learning Algorithm for Continually Running Fully Recurrent Neural Networks," *ICS Report 8805*, Institute for cognitive Science, University of California, San Diego, 1988.

51. Williams, R. J., and Zipser, D., "Experimental Analysis of the Real-time Recurrent Learning Algorithm," *Connection Science*, Vol. 1, No. 1, pp. 87, 1989.

52. Wolf, L., "Recurrent Nets for the Storage of Cyclic Sequences," *Proceedings of 1988 IEEE International Conference on Neural Networks*, San Diego, 1988, I-53.

53. Wong, A. K. C., and Vieth, J. O., "Neural Networks in Statistical Classification," *Proceedings of the International Joint Conference on Neural Networks*, Washington, D. C., January 1990, I-553.

54. Wu, X., "Material Modeling: An Approach with Neural Networks and Computational Mechanics," *Ph. D. thesis proposal*, Department of Civil Engineering, University of Illinois at Urbana-Champaign, March 1990.

2.7 Unsupervised Learning Systems

Unsupervised learning or self-organizing systems are primarily inspired by our understanding on human information process from extensive research in the psychological and biological processes of cognition. In unsupervised learning, the network is not taught the regularity and relations in the set of training patterns, instead the network captures the regularities of the input vectors by using unsupervised learning procedures. This phenomenon is remarkable because it provides a computational model with natural resemblance to the cognitive process of human beings. The capability of self-organization also makes this kind of network a powerful tool for real-time pattern classification and signal processing applications where the target classifications are not known a priori yet the data can be sorted into different categories. On the other hand, from stability theory concerning the computational properties of self-organizing networks, the stabilization of the system has a definite relation with the minimization of an energy measurement of the system so that self-organizing systems have great potential in solving constraint optimization problems. To date, Hopfield's network (Hopfield, 1982) and Kohonen's self-organizing network (Kohonen, 1984) have been successfully used for some combinatorial optimization problems in different fields, including the traveling salesman problem (TSP).

In unsupervised learning, self-organization is imparted via learning rules based on Hebbian learning (Hebb, 1949) and Competitive learning (Grossberg, 1976; Rumelhart and Zipser, 1986) or variants of both. In Hebbian learning, the modification of weights is based on the correlation between the presynaptic and postsynaptic activity of a neuron. The weight of connection is increased if the correlation is positive (excitation), otherwise the weight of connection is decreased. Competitive learning is a pattern classification procedure for conditioning intra-layer connections in a two layer network such that the input vectors are properly classified into distinct clusters. Competition and inhibition are two basic mechanisms that provide dynamics to the system. For example, competitive layers and inhibitory connections are salient features of the Kohonen network, the Counterpropagation network (Hecht-Nielsen, 1987) and Adaptive Resonance Theory (Grossberg, 1976; Carpenter and Grossberg, 1986).

In this section, a brief overview is provided on some widely used unsupervised learning models and some recent developments in the area. Note that the reference listing does not necessarily include the early work on the analysis of different learning paradigms and such information can be found in numerous books recently published on neural networks.

2.7.1 Hebbian Learning Rule

In self-organizing systems, most of the learning rules for modifying the connection strengths of existing connections are a variant of Hebbian learning. Hebbian learning is a correlation rule based on observations from physiological and psychological studies on cognition. In his book, Organization of Behavior (1949), Hebb states that:

> *When an axon of cell A is near enough to excite a cell B and repeatedly or persistently takes part in firing it, some growth process or metabolic change takes place in one or both cells such that A's efficiency as one of the cells firing B is increased.*

In simple terms, we can state Hebb's rule as: if a unit receives an input from another unit and both are active, the connection weight between these two units should be increased. This simple learning rule is usually put in the following mathematical form:

$$\Delta w_{ij} = \eta \; a_i \; a_j$$

where η is the learning rate, a_i and a_j are activations of both units, and Δw_{ij} is the modification on the connection weight.

One disadvantage of using this simple correlation learning rule is that it is not goal bounded like that of a delta rule. Different versions of the Hebbian style learning rule have been proposed by many researchers. Sejnowski (1977) uses the covariance correlation to replace the simple correlation such that

$$\Delta w_{ij} = \eta \; (a_i - \overline{a_i}) \; (a_j - \overline{a_j})$$

where $\overline{a_i}$ is the mean value. Sutton and Barto (1981) use the correlation of the mean value and the variance in the learning rule in the following form:

$$\Delta w_{ij} = \eta \; (\overline{a_i}) \; (a_j - \overline{a_j})$$

Klopf (1986) introduced the drive-reinforcement learning by using the correlation in the changes of activation such that

$$\Delta w_{ij} = \eta \; \Delta a_i \; \Delta a_j$$

Of course, combinations of the above schemes also produce some new schemes, such as the one proposed by Cheung and Omidvar (1988):

$$\Delta w_{ij} = \eta \; a_i \; w_{ij} \; \Delta a_j$$

A major improvement on Hebbian learning is the introduction of decaying effect, which has been illustrated by Grossberg (1968), Hopfield (1984) and others. The learning rule is:

$$\Delta w_{ij} = -w_{ij} + F(a_i) \; F(a_j)$$

in which F is the activation function. Based on this, Kosko (1986) proposed his differential Hebbian learning in the following way:

$$\Delta w_{ij} = -w_{ij} + F'(a_i) \; F'(a_j)$$

where F′ is the derivative of F with respect to the activation value.

2.7.2 The Competitive Learning Architecture

The simplest form of competitive learning networks consists of two layers; the input layer for receiving input patterns and the competitive layer for classifying the input vectors (Rumelhart and Zipser, 1986). The weights are usually limited in the neighborhood of (0, 1), and the sum of weights to a unit

is always 1. The competition in the competitive layer is accomplished through a winner-takes-all scheme, in which the unit with the highest sum of weights is assigned as the winner. The activation of the winner is then set at 1.0 and the remaining units are given the value of 0.0. The winner-takes-all scheme can be represented in the following form:

$$a_j = \begin{cases} 1 \text{ if } N_j \ (= \sum a_j w_{ji}) > N_i, \quad \text{for all } i, \ i \neq j \\ 0 \text{ otherwise} \end{cases}$$

where a_j is the activation value of unit i. The weight updating is made after the winner is selected, and only the weights that correspond to the connections to the winner are updated in the following form:

$$\Delta w_{ij} = \eta \left(\frac{a_i}{n} - w_{ij}\right)$$

in which, η is the learning rate, and n is the number of units in the input layer that have activation levels of 1.0. The simplistic architecture of the network is shown in Fig. 2.7.

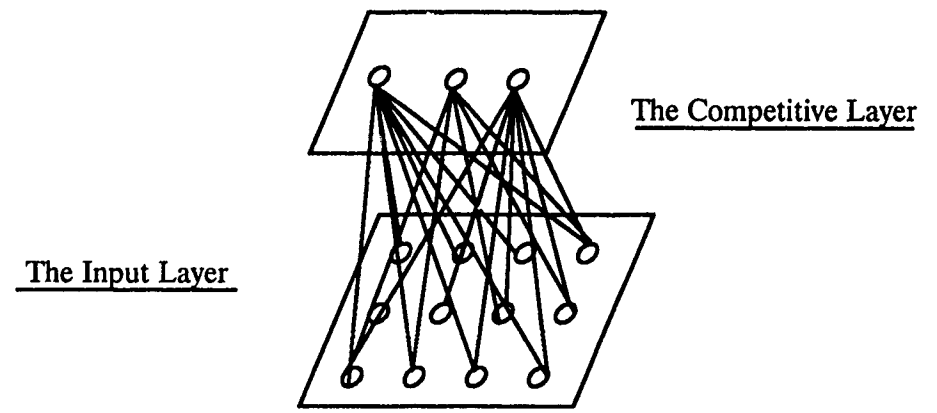


Figure 2.7 – A Simple Architecture of a Competitive Learning Network

It should be pointed out that the competitive layer can also be implemented with inhibition connections instead of the simple winner-takes-all scheme. With full or lateral inhibition connections, the activation levels of the processing units gradually relax to the point where the unit with the highest incoming sum remains activated so that it is chosen as the winner. Such a system would be more biologically plausible.

2.7.3 The Hopfield Network

Hopfield introduced the binary version of the network in 1982 and later extended it to treat analog values in 1984. The basic structure and operation of the two versions of the Hopfield network is essentially the same. For simplicity, the binary version is described here.

The Hopfield network is a single layer network in which each unit is connected to every other unit. The network is recursive because the output of each unit feed into inputs of other units in the same layer. The weight matrix is symmetric such that the network is able to converge to a stable state. Each unit in a Hopfield network has a binary activation value or state, that is, one of the two binary states. The state of the network at a moment is represented by a state vector containing the activation value of each unit. The state of the network can be changed over time until finally it settles to a stable state; at this moment, the corresponding energy measurement of the network reaches its optimum value which is usually a local minimum of the energy function. To find the global minimum of the energy function that the system represents or corresponds to, a restarting scheme or the use of Boltzmann and Cauchy machines is needed.

There are two processes involved with the self-organizing process of a Hopfield network, namely, the setting of connection weights and the state vector updating. The connection strengths or weights are usually wired instead of through training. After the state vector is initialized, the updating of state vector proceeds in a very simple procedure. For each neuron, calculate the weighted sum of its inputs. If the sum is larger than or equal to zero, then change the activation of the unit to 1.0; otherwise set the activation value to 0. Selection of the next unit for updating can be done sequentially or randomly. This process continues through all the units in the network until a stable state of the network is reached. The architecture of a Hopfield network is shown in Fig. 2.8.

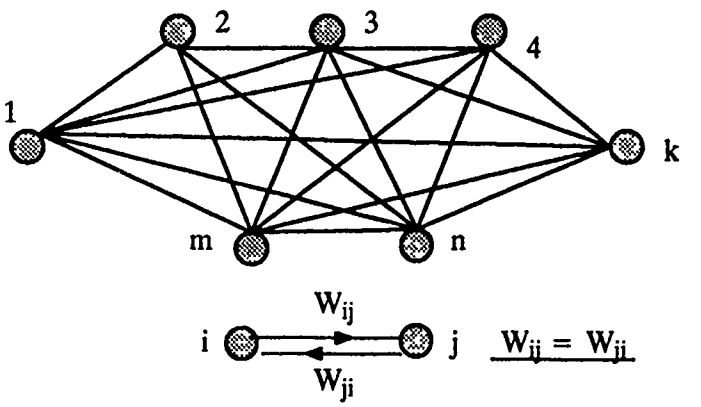


Figure 2.8 – A Simple Architecture of the Hopfield Network

2.7.4 The Kohonen Self-Organizing Network

The Kohonen self-organizing network is specially designed by Kohonen (1984) for regularity determination and feature extraction in the input patterns. Like the competitive learning network, the Kohonen network usually comes in two layers, an input layer for receiving input patterns and a competitive layer processing the input information. Input patterns are classified by the units that they activate in the competitive layer and the activation patterns of the competitive layer represent the identification of the network. The competitive layer is commonly organized as a two dimensional grid. Full connec-

tions from the input layer to the competitive layer are enforced and the connection strengths or weights are initialized with random values. The general architecture of a Kohonen network is shown in Fig. 2.9.

Though the architecture of a Kohonen network is similar to that of a competitive learning architecture, the self-organizing process or learning in the competitive layer is accomplished via a different criterion. Firstly, a matching value that measures the closeness of a weight vector of each unit in the competitive layer with the corresponding vector of input pattern is calculated by

$$\| \mathbf{P} - \mathbf{W}_i \| = [\sum_j (p_j - w_{ij})^2]^{\frac{1}{2}}$$

in which, **P** is the input vector containing the activations of the input nodes, and $\mathbf{W}_i$ is the weight vector of unit i in the competitive layer with connections to all the nodes in the input layer. The winning node is identified with the minimum matching value. After identifying the winning node, the next step is to select the neighborhood of nodes around the winner for weight updating. Only the weights of those nodes in the winning neighborhood are modified with the following equation:

$$\Delta w_{ij} = \begin{cases} \alpha(p_j - w_{ij}) & \text{if unit i is in the winning neighborhood} \\ 0 & \text{otherwise} \end{cases}$$

where α is the learning rate. Usually, the learning rate decreases as the training proceeds. On the other hand, the size of the winning neighborhood in the competitive layer can be given a relative large width initially, and then reduce the size with further training.

It is clear that a Kohonen network performs a feature mapping between the input pattern and the representing weight vector and identifies the input pattern with activation pattern in the competitive layer. The feature mapping capability makes this network suitable for applications in sensory motor control, language processing and constraint optimization.

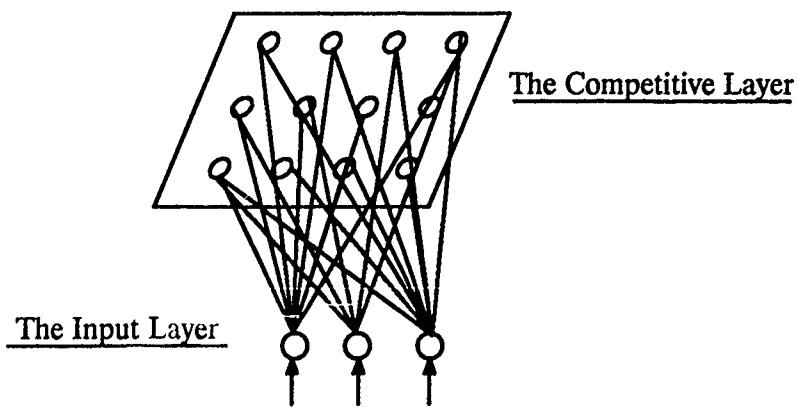


Figure 2.9 – A Simple Architecture of the Kohonen Network

2.7.5 Other Unsupervised Learning Models

There are many other unsupervised learning models such as ART – the Adaptive Resonance Theory proposed by Grossberg and Carpenter (1976, 1987, 1988, 1990), which is a two layer, nearest-neighbor classifier that stores an arbitrary number of spatial patterns using competitive learning; BAM – the Bidirectional Associative Memory introduced by Kosko (1987, 1988), using Hebbian learning to encode arbitrary spatial pattern pairs in a two layer, heteroassociative pattern matcher; and FAM – Fuzzy Associative Memory also proposed by Kosko (1987). Another interesting architecture is Hecht-Nielsen's Counterpropagation Network (1987). A counterpropagation network is actually a hybrid three layer network, in which the hidden layer is a Kohonen competitive layer with unsupervised learning and the rest are standard backpropagation layers trained with the Generalized Delta Rule. Due to time limitation, we cannot describe all the other networks in this survey. The following list of references will provide pointers to recent development in unsupervised learning paradigms and related works.

General References

1. Abe, S., "Theories on the Hopfield Neural Networks," *Proceedings of the International Joint Conference on Neural Networks*, Washington, D. C., 1989, I-557.

2. Amari, S., "Statistical Neurodynamics of Various Versions of Correlation Associative Memory," *Proceedings of 1988 IEEE International Conference on Neural Networks*, I-633.

3. Asogawa, M., "Adaptive Input Field Neural Network," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17-21, 1990, III-733.

4. Bak, C. S., and Little, M. J., "Memory Capacity of Artificial Neural Networks with High Order Node Connections," *Proceedings of 1988 IEEE International Conference on Neural Networks*, I-207.

5. Banzhaf, W., "Towards Continuous Models of Memory," *Proceedings of the First IEEE International Conference on Neural Networks*, San Diego, 1987, II-223.

6. Baum, E. B., Moody, J., and Wilczek, F., "New Neural Algorithms for Associative Memory," *Proceedings of the First IEEE International Conference on Neural Networks*, San Diego, 1987, II-505.

7. Bhatti, A. A., "A Neural Network Based Multi-Associative Memory Model," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17-21, 1990, I-893.

8. Carpenter, G., and Grossberg, S., "Adaptive Resonance Theory: Stable Self-organization of Neural Recognition Codes in Response to Arbitrary Lists of Input Patterns," *Eighth Annual Conference of the Cognitive Science Society*, Lawrence Erlbaum Associates, 1986.

9. Carpenter, G., and Grossberg, S., "ART 2: Self-organization of Stable Category Recognition Codes for Analog Input Patterns," *Proceedings of the IEEE International Conference on Neural Networks*, 1987, II-727.

10. Carpenter, G., and Grossberg, S., "ART 3: Hierarchical Search Using Chemical Transmitters in Self-Organizing Pattern Recognition Architectures," *Neural Networks*, Vol 3, pp. 129–152, 1990.

11. Carpenter, G. A., and Grossberg, S., "Search Mechanisms for Adaptive Resonance Theory (ART) Architecture," *Proceedings of the International Joint Conference on Neural Networks*, Washington, D. C., 1989, I–201.

12. Chen, S.–S., "Automated Reasoning on Neural Networks: A Probabilistic Approach," *Proceedings of the First IEEE International Conference on Neural Networks*, San Diego, 1987, II–373.

13. Cheung, J., and Omidvar, M., "Mathematical Analysis of Learning Behavior of Neuronal Models," in *Neural Information Processing systems – Natural and Synthetic*, D. Anderson, Ed., American Institute of Physics, pp. 174–183, 1988.

14. Chiueh, T., and Goodman, R. M., "High-Capacity Exponential Associative Memories," *Proceedings of 1988 IEEE International Conference on Neural Networks*, I–153.

15. Cohen, M. A., "Sustained Oscillation in Symmetric Cooperative–Competitive Neural Networks," *Proceedings of the First IEEE International Conference on Neural Networks*, San Diego, 1987, II–127.

16. Cruz, C. A., Hanson, W. A., and Tam, J. Y., "Knowledge Processing Through Flow–of–Action," *Proceedings of the First IEEE International Conference on Neural Networks*, San Diego, 1987, II–343.

17. Dajani, A. L., Kamel, M., and Elmasry, M. I., "Single Layer Potential Function Neural Network for Unsupervised Learning," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17–21, 1990, II–273.

18. Darken, C., and Moody, J., "Fast Adaptive K–Means Clustering: Some Empirical Results," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17–21, 1990, II–233.

19. Davis, G., and Ansari, A., "Sensitivity Analysis of Hopfield Net," *Proceedings of the First IEEE International Conference on Neural Networks*, San Diego, 1987, III–325.

20. Dayhoff, J. E., "Graphical Folding and Distributed Failures in Large Network Simulations," *Proceedings of the First IEEE International Conference on Neural Networks*, San Diego, 1987, II–231.

21. DeSieno, D., "Adding A Conscience to Competitive Learning," *Proceedings of 1988 IEEE International Conference on Neural Networks*, I–117.

22. Fiesler, E., Caulfield, H. J., Choudry, A., and Ryan, J. P., "Maximum Capacity Topologies for Fully Connected Layered Neural Networks with Bidirectional Connections," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17–21, 1990, I–827.

23. Floreen, P., "An Analysis of the Convergence Time of Hamming Memory Networks," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17–21, 1990, I–867.

24. Fuchs, A., and Haken, H., "Pattern Recognition and Associative Memory as Dynamical Processes in Nonlinear Systems," *Proceedings of 1988 IEEE International Conference on Neural Networks*, I-217.

25. Gaines, B. R., "Positive Feedback Processes Underlying Functional Differentiation," *Proceedings of the First IEEE International Conference on Neural Networks*, San Diego, 1987, II-387.

26. Gallant, C. C., and Hinton, G. E., "Discovering High Order Features with Mean Field Modules," in *Advances in Neural Information Processing Systems II*, D. S. Touretzky, Ed., Morgan Kaufmann Publishers, San Mateo, CA, pp. 509-516, 1990.

27. Geffner, H., and Pearl, J., "On the Probabilistic Semantics of Connectionist Networks," *Proceedings of the First IEEE International Conference on Neural Networks*, San Diego, 1987, II-187.

28. Geiger, D., and Girosi, F., "Coupled Markov Random Fields and Mean Field Theory," in *Advances in Neural Information Processing Systems II*, D. S. Touretzky, Ed., Morgan Kaufmann Publishers, San Mateo, CA, pp. 661-667, 1990.

29. Geva, S., and Sitte, J., "A Pseudo-Inverse Neural Net with Storage Capacity Exceeding N," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17-21, 1990, I-783.

30. Green, P. E., and Michalson, W. R., "Real-Time Evidential Reasoning and Network Based Processing," *Proceedings of the First IEEE International Conference on Neural Networks*, San Diego, 1987, II-359.

31. Grossberg, S., "Some Nonlinear Networks Capable of Learning a Spatial Pattern of Arbitrary Complexity," *Proceedings of the National Academy of Sciences*, 59, pp. 368-372, 1968.

32. Grossberg, S., "Adaptive Pattern Classification and Universal Recoding: Part 1. Parallel Development and Coding of Natural Features Detectors," *Biological Cybernetics*, 23, pp. 121-134, 1976.

33. Grossberg, S., "Self-Organizing Neural Architectures for Motion Perception, adaptive sensory-motion control and associative mapping," *Proceedings of the International Joint Conference on Neural Networks*, Washington, D. C., 1990, II-26.

34. Haghighi, S., and Akers, L. A., "TRELIS: An Optimal Unsupervised Local Training Rule," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17-21, 1990, II-349.

35. Haines, K., and Hecht-Nielsen, R., "A BAM with Increased Information Storage Capacity," *Proceedings of 1988 IEEE International Conference on Neural Networks*, I-181.

36. Hassoun, M. H., Song, J., Shen, S.-M., and Spitzer, A. R., "Self-Organizing Autoassociative Dynamic Multiple-Layer Neural Net for the Decomposition of Repetitive Superimposed Signals," *Proceedings of the International Joint Conference on Neural Networks*, Washington, D. C., 1990, I-621.

37. Hassoun, M. H., and Youssef, A. M., "Autoassociative Neural Memory Capacity and Dynamics," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17-21, 1990, I-763.

38. Hebb, D. O., *The Organization of Behavior: A Neuropsychological Theory*, Wiley, New York, 1949.

39. Hecht-Nielsen, R., "Counterpropagation Networks," *Applied Optics*, 26, pp. 4979–4984, 1987.

40. Hirai, Y., "Representation of Control Structures by a Model of An Associative Processor, HASP," *Proceedings of the First IEEE International Conference on Neural Networks*, San Diego, 1987, II–447.

41. Hirsch, M. W., "Convergence in Neural Nets," *Proceedings of the First IEEE International Conference on Neural Networks*, San Diego, 1987, II–115.

42. Hodges, R. E., and Wu, C.–H., "A Method to Establish an Autonomous Self–organizing Feature Map," *Proceedings of the International Joint Conference on Neural Networks*, Washington, D. C., 1990, I–517.

43. Hollatz, J., and Schurmann, B., "The Detailed Balance Net: A Stable Asymmetric Artificial Neural System for Unsupervised Learning," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17–21, 1990, III–453.

44. Hopfield, J., "Neural Networks and Physical Systems with Emergent Collective Computational Abilities," *Proceedings of the National Academy of Sciences*, 79, pp. 2554–2558, 1982.

45. Hopfield, J., "Neurons with Graded Response Have Collective Computational Properties Like Those of Two–State Neurons," *Proceedings of the National Academy of Sciences*, 81, pp. 3088–3092, 1984.

46. Hopfield, J., and Tank, D., "Neural Computation of Decisions in Optimization Problems," *Biological Cybernetics*, 52, pp. 141–152, 1985.

47. Hopfield, J., and Tank, D., "Computing with Neural Circuits: A Model," Science, 233, pp. 625–633, 1986.

48. Hrycej, T., "Self–Organization by Delta Rule," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17–21, 1990, II–307.

49. Humpert, B., "Bidirectional Associative Memory with Several Patterns," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17–21, 1990, I–741.

50. Intrator, N., "A Neural Network for Feature Extraction," in *Advances in Neural Information Processing Systems II*, D. S. Touretzky, Ed., Morgan Kaufmann Publishers, San Mateo, CA, pp. 719–726, 1990.

51. Jagota, A., and Jakubowicz, O., "Knowledge Representation in a Multi–layered Hopfield Network," *Proceedings of the International Joint Conference on Neural Networks*, Washington, D. C., 1989, I–435.

52. Jeong, H., and Park, J. H., "Lower Bounds of Annealing Schedule for Boltzmann and Cauchy Machines," *Proceedings of the International Joint Conference on Neural Networks*, Washington, D. C., 1989, I–581.

53. Kan, W.–K., and Aleksander, I., "A Probabilistic Logic Neuron Network for Associative Learning," *Proceedings of the First IEEE International Conference on Neural Networks*, San Diego, 1987, II–541.

54. Kangas, J., "Time-Delayed Self-Organizing Maps," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17-21, 1990, II-331.

55. Kennedy, M. P., and Chua, L. O., "Circuit Theoretic Solutions for Neural Networks - An Old Approach to a New Problem," *Proceedings of the First IEEE International Conference on Neural Networks*, San Diego, 1987, II-169, I-435.

56. Klopf, A., "Drive-reinforcement Model of Single Neuron Function: An Alternative to the Hebbian Neuronal Model," in *Neural Networks for Computing: AIP Conference Proceedings 151*, J. Denker, Ed., American Institute of Physics, pp. 265-270, 1986.

57. Kohonen, T., *Self-Organization and Associative Memory*, Springer-Verlag, Berlin, 1984.

58. Kohonen, T., "Learning Vector Quantization," *Neural Networks Supplement: INNS Abstracts*, 1, pp. 303, 1988.

59. Kohonen, T., Barna, G., and Chrisley, R., "Statistical Pattern Recognition with Neural Networks: Benchmarking Studies," *Proceedings of 1988 IEEE International Conference on Neural Networks*, I-61.

60. Koikkalainen, P., and Oja, E., "Self-organizing Hierarchical Feature Maps," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17-21, 1990, II-279.

61. Kosko, B., "Fuzzy Knowledge Combination," *International Journal for Intelligent Systems*, 1, pp. 293-320, 1986.

62. Kosko, B., "Adaptive Inference in Fuzzy K nowledge Networks," *Proceedings of the First IEEE International Conference on Neural Networks*, San Diego, 1987, II-261.

63. Kosko, B., "Competitive Adaptive Bidirectional Associative Memories," *Proceedings of the First IEEE International Conference on Neural Networks*, San Diego, 1987, II-759.

64. Kosko, B., "Feedback Stability and Unsupervised Learning," *Proceedings of 1988 IEEE International Conference on Neural Networks*, San Diego, 1988, I-141.

65. Kosko, B., "Unsupervised Learning in Noise," *Proceedings of the International Joint Conference on Neural Networks*, Washington, D. C., 1989, I-7.

66. Kosko, B., "Stochastic Competitive Learning," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17-21, 1990, II-215.

67. Kuczewski, R. M., Myers, M. H., and Crowford, W. J., "Exploration of Backward Error Propagation as A Self-Organized Structure," *Proceedings of the First IEEE International Conference on Neural Networks*, San Diego, 1987, II-89.

68. Kurten, K. E., "Phase Transitions in Quasi-random Neural Networks," *Proceedings of the First IEEE International Conference on Neural Networks*, San Diego, 1987, II-197.

69. Laskoski, G. M., "Addressing Scaling and Plasticity Problems with a Biologically Motivated Self-organizing Network," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17-21, 1990, II-355.

70. Lee, B. W., and Sheu, B. J., "An Investigation on Local Minima of Hopfield Network for Optimization Circuits," *Proceedings of 1988 IEEE International Conference on Neural Networks*, I-45.

71. Lee, S., and Kil, R. M., "Bidirectional Continuous Associator Based on Gaussian Potential Function Network," *Proceedings of the International Joint Conference on Neural Networks*, Washington, D. C., 1989, I-45.

72. Lehar, S., Weaver, J., "A Developmental Approach to Neural Network Design," *Proceedings of the First IEEE International Conference on Neural Networks*, San Diego, 1987, II-97.

73. Lemmon, M., and Kumar, BVK V., "Neural Networks as Associate Memories," *Proceedings of the First IEEE International Conference on Neural Networks*, San Diego, 1987, II-213.

74. Lemmon, M., and Kumar, BVK V., "Learning in Competitively Inhibited Neural Nets," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17-21, 1990, III-477.

75. Lenz, R., and Osterberg, M., "Leaning Filter Systems," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17-21, 1990, II-337.

76. Loos, H. G., "Quadratic Hadamard Memories," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17-21, 1990.

77. Luttrell. S. P., "Self-Organizing Multilayer Topographic Mappings," *Proceedings of 1988 IEEE International Conference on Neural Networks*, I-93.

78. Macukow, B., and Arsenault, H. H., "Neural Network Model Using A Normalized Inner Product as A Measure of Similarity," *Proceedings of 1988 IEEE International Conference on Neural Networks*, I-225.

79. Martland, D., "Behavior of Autonomous, Synchronous, Boolean Networks," *Proceedings of the First IEEE International Conference on Neural Networks*, San Diego, 1987, II-243.

80. Matsuyama, Y., "Multiple Descent Cost Competitive Learning: Batch and Successive Self-organization with Excitatory and Inhibitory Connection," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17-21, 1990, II-299.

81. Maxwell, T., Giles, C. L., and Lee, Y. C., "Generalization in Neural Networks: The Contiguity Problem" *Proceedings of the First IEEE International Conference on Neural Networks*, San Diego, 1987, II-41.

82. Mjolsness, E., "Control of Attention in Neural Networks," *Proceedings of the First IEEE International Conference on Neural Networks*, San Diego, 1987, II-567.

83. Moody, J., "Perspectives on Associative Memories," *Proceedings of the First IEEE International Joint Conference on Neural Networks*, San Diego, 1987, III-59.

84. Moody, J., and Darken, C., "Learning with Localized Receptive Fields," *Proceedings of the 1988 Summer School Connectionist Models*, D. Touretzky, G. Hinton and T. Sejnowski, Eds., pp. 133-143, Carnegie Mellon University, June 17-26, 1988.

85. Moody, J., "Dynamics of Lateral Interaction Networks," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17-21, 1990, III-483.

86. Moore, B., "ART 1 and Pattern Clustering," *Proceedings of the 1988 Summer School Connectionist Models*, D. Touretzky, G. Hinton and T. Sejnowski, Eds., June, 17-26, 1988, pp. 175-185, Carnegie Mellon University.

87. Nagashino, H., Kinouchi, Y., and Ushita, T., "Self-Organizing Process of a Neural Network Model for Pattern Detection," *Proceedings of the First IEEE International Conference on Neural Networks*, San Diego, 1987, II-105.

88. Ngee, T. H., and Hock, O. T., "A Novel generalized Flip Flop for Memory Association and Maximization in Artificial Neural Network," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17-21, 1990, I-885.

89. Nigrin, A. L., "SONNET: A Self-Organizing Neural Network that Classifies Multiple Patterns Simultaneously," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17-21, 1990, II-313.

90. Ntourntoufis, P., "Self-organizing Properties of a Discriminator-based Neural Network," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17-21, 1990, II-319.

91. Okajima, K., Tanaka, S., and Fujiwara, S., "A Heteroassociative Memory Network with Feedback Connection," *Proceedings of the First IEEE International Conference on Neural Networks*, San Diego, 1987, II-711.

92. Plumbley, M. D., and Fallside, F., "An Information-Theoretic Approach to Unsupervised Connectionist Models," *Proceedings of the 1988 Summer School Connectionist Models*, D. Touretzky, G. Hinton and T. Sejnowski, Eds., pp. 239-245, Carnegie Mellon University, June 17-26, 1988.

93. Reggia, J. A., "Properties of A Competition-based Activation Mechanism In Neuromimetic Network Models," *Proceedings of the First IEEE International Conference on Neural Networks*, San Diego, 1987, II-131.

94. Reggia, J. A., "Methods for Deriving Competitive Activation Mechanisms," *Proceedings of the International Joint Conference on Neural Networks*, Washington, D. C., 1989, I-357.

95. Richfield, S., "High-Speed Learning in Deductive-Reasoning Systems," *Proceedings of the First IEEE International Conference on Neural Networks*, San Diego, 1987, II-575.

96. Ritter, H., and Schulten, K., "Kohonen's Self-Organizing Maps: Exploring their Computational Capabilities," *Proceedings of 1988 IEEE International Conference on Neural Networks*, I-109.

97. Rumelhart, D. E., and Zipser, D., "Feature discovery by Competitive Learning," in *Parallel Distributed Processing*, D. E. Rumelhart, J. L McClelland and PDP Group, The MIT Press, 1986.

98. Salam, F. M. A., Bai, S., and Hou, J., "Dynamics of Feedback Neural Nets with Unsupervised Learning," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17-21, 1990, II-239.

99. Saerens, M., "Unsupervised Clustering Based on A Competitive Cost Function," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17-21, 1990, II-343.

100. Samad, T., and Harper, P., "Associative Memory Storage Using a Variant of the Generalized Delta Rule," *Proceedings of the First IEEE International Joint Conference on Neural Networks*, San Diego, 1987, III-173.

101. Sanger, T. D., "An Optimality Principle for Unsupervised Learning," in *Advances in Neural Information Processing Systems I*, D. S. Touretzky, Ed., Morgan Kaufmann Publishers, San Mateo, CA, 1989, 11-19.

102. Schurmann, B., "Generalized Adaptive Bidirectional Associative Memory," *Proceedings of the International Joint Conference on Neural Networks*, Washington, D. C., 1989, I-91.

103. Schyns, P. G., "Expertise Acquisition Through Concepts Refinement in A Self-organizing Architecture," *Proceedings of the International Joint Conference on Neural Networks*, Washington, D. C., 1990, I-236.

104. Seidman, S. B., Reggia, J. A., and Wang, P. Y., "Analysis of a Competition-based Connectionist Model," *Proceedings of the First IEEE International Conference on Neural Networks*, San Diego, 1987, II-139.

105. Sejnowski, T., " Storing Covariance with Nonlinear Interacting Neurons," Journal of Mathematical Biology, 64, pp. 385-389, 1977.

106. Shiue, L. C., and Grondin, R. O., "On Designing Fuzzy Learning Neural-Automata," *Proceedings of the First IEEE International Conference on Neural Networks*, San Diego, 1987, II-299.

107. Steels, L., "Self-Organization Through Selection," *Proceedings of the First IEEE International Conference on Neural Networks*, San Diego, 1987, II-55.

108. Stotzka, R., and Manner, R., "Self-organization of a Linear Multilayered Feedforward Neural Network," *Proceedings of the International Joint Conference on Neural Networks*, Washington, D. C., 1990, I-126.

109. Sudharsanan, S. I., and Sundareshan, M. K., "Systematic Design of Associative Memory Networks: Equilibrium Confinement, Exponential Stability and Gradient Descent Learning," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17-21, 1990, I-757.

110. Sung, C. H., and Wilson, D., "Percognitron, Neocognitron Coupled with Perceptron," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17-21, 1990, III-753.

111. Sutton, R., and Barto, A., "Toward a Modern Theory of Adaptive Networks: Expectation and Prediction," *Psychological Review*, 88, pp. 135-171, 1981.

112. Sutton, G. G., Reggia, J. A., and Maisog, J. M., "Competitive Learning Using Competitive Activation Rules," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17-21, 1990, II-285.

113. Tan, S., Vandenberghe, L., Vandewalle, J., "Remarks on the Stability of Asymmetric Dynamical Neural Networks," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17-21, 1990, III-461.

114. Tillery, S., and Combs, N., "The Problem of Internal Representations in Neural Nets: Concepts, Implications, and A Proposed Metric," *Proceedings of the First IEEE International Conference on Neural Networks*, San Diego, 1987, II-585.

115. Tsai, W. K., Palos, A., and Fernandez, B., "ASDM–A Novel Neural Network Model Based on Sparse Distributed Memory," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17–21, 1990, I–771.

116. Tsutsumi, K., "Cross-Coupled Hopfield Nets via Generalized Delta Rule Based Internetworks," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17–21, 1990, II–259.

117. Wang, Y.–F., Cruz Jr., J. B., and Mulligan Jr., J. H., "An Enhanced Bidirectional Associative Memory," *Proceedings of the International Joint Conference on Neural Networks*, Washington, D. C., 1989, I–105.

118. Yoshimoto, S., "A Study on Artificial Neural Network Generalization Capability," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17–21, 1990, III–689.

119. Zak, M., and Toomarian, N., "Unsupervised Learning in Neurodynamics Using the Phase Velocity Field Approach," in *Advances in Neural Information Processing Systems II*, D. S. Touretzky, Ed., Morgan Kaufmann Publishers, San Mateo, CA, pp. 583–589, 1990.

3 ENGINEERING APPLICATIONS

3.1 General

To date, the resurgence of research in neural networks not only has advanced the technology in some branches of networks and resulted in sophisticated modeling tools but also has generated various applications in many disciplines. In addition to intensive research and application in the traditional areas such as the modeling of cognitive process, pattern recognition, and language processing, a strong trend is seen in the application of neural networks to real engineering problems. As a by-product of this endeavor, many innovative engineering approaches have also been introduced to the development of neural network based modeling systems.

Because of the characteristics associated with neural network modeling, there are several kinds of engineering problems that are suitable for this technology. As a knowledge representation tool, neural networks can be used in modeling the behavior of engineering material in structural engineering and computational mechanics, system identification, control, and prediction. As a computational tool, Hopfield-type networks have found extensive applications in planning, scheduling, and optimization. As a classifier and pattern matcher, neural networks can provide an alternative in solving problems associated with pattern recognition, diagnostics, maintenance, and image processing.

It should be realized that neural networks are only a tool for some specially suited problems, and there is still a long way to go before this technology becomes a sophisticated entity in the tool box for engineers. The current application of neural networks is more of an art than a science because the implementation process involves a lot of heuristics and engineering judgement. The success of application, in some ways, depends on the modeler's understanding of the problem and the selection of certain parameters in using a neural network. At this stage, the most effective use of this technology may rest on the development of systems combining a traditional approach with this technology.

In the following paragraphs, some of the applications in different engineering fields are described and relevant references provided. As it has been noted before, the reference list is by no means exhaustive and we owe an apology to those researchers whose work has been overlooked due to time limits on preparing this report.

3.2 Hybrid Systems and Their Applications

As mentioned before, neural networks reside on the middle ground between a pure mathematically based engineering approach and the symbolic dominant AI approach. A system combining the computational capability of neural networks and the deep reasoning ability of KBES (Knowledge-based Expert Systems) would most likely offer new insight and powerful tools in engineering problem solving. The objective of building hybrid expert systems, that is, integrating neural networks and expert systems, is to explore the advantages and neutralize the disadvantages of both systems.

In a hybrid expert system, neural networks usually function as a classifier for data evaluation, regularity detection and classification or as an optimizer for solving a multiconstraint optimization prob-

lem in KBES. In the last few years, hybrid systems have been successfully constructed in planning, scheduling, diagnosis, and decision making. In general, as summarized by Rabelo, Alptekin and Kiran (1990), the integration of Neural Networks and KBES takes these forms: 1) Neural Networks are used for knowledge representation, and the represented knowledge is then translated to rules to a KBES for symbolic manipulation, 2) a KBES is used to obtain a preliminary solution, which is then optimized by Neural Networks, and 3) Neural Networks are used within a KBES to perform tasks that explicit rules would be too complex to build.

There are, of course, many more variations on the design of a hybrid system. In the following paragraphs, some of the typical approaches on the construction of a hybrid system are described and an overview of their applications as well as references on the development of Connectionist Expert Systems (CES) are also provided. It should be pointed out that CES are different from the simple integration of KBES with neural networks, rather that CES are standalone systems capable of rule extraction and generalization within themselves (Gallant, 1988). Some research works on novel approaches to rule extraction from connectionist systems are also included for reference purpose.

3.2.1 The Three-Stage Integration – Hillman (1990)

According to Hillman (1990), three stages are involved in building a hybrid expert system:

1. The data and information obtained are preprocessed by the expert system;

2. The preprocessed data are filtered through a neural networks for evaluation, regularity detection and classifications;

3. Results from the neural network are analyzed and synthesized by the expert system.

The advantage of using a neural network as a data evaluator and regularity detector is the simplification of the rule building process in data evaluation, reducing the execution time. It is apparent that

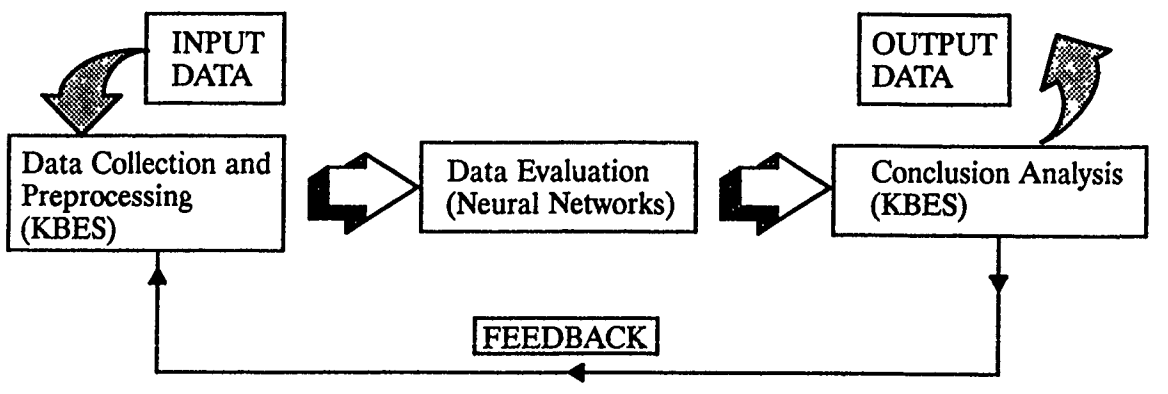


Figure 3.1 -- The Three-Stage Approach to the Hybrid KBES

software tools for interface are of primary importance in the integration process of the hybrid system. In Hillman's toy problem, two commercially available packages - AUBREY and NeuroShell (Ward Systems Group, Inc.), were used. This approach is illustrated in Fig. 3.1.

3.2.2 Two-Stage Integration - Benachenhou, et al. (1990)

Similar to the three-stage approach, the two-stage approach usually takes the following forms: 1) a neural network works as a preprocessor for an expert system, and 2) another neural network works as a postprocessor on results from the expert system.

In this paper, a hybrid system consisting of a feature-based knowledge system and an ART1 (Carpenter and Grossberg, 1987) network is used for the inverse problem of image processing, that is, sorting images by clustering instead of extracting features from images by clustering in the direct image processing problem. The knowledge based system provides a training environment by giving a few known features of different images and the ART1 network then sorts those images into unknown numbers of classes by using its clustering capability. The advantage of using ART1 is that it can work on an open set of samples, whereas for the Kohonen network the number of clustering groups should be known *a priori*. Thus the system can be used to classify unfamiliar images into new classes.

The system is applied to clustering a small set of primers among a large open set generated by a rule based system and uses the sorted results in the diagnosis of AIDS virus-mutated DNA by a recombinant DNA technology called Polymerase Chain Reaction (PCR). In the test results, the hybrid architecture was able to select the leaders of image clustering, and the system is currently evaluated under practical medical conditions.

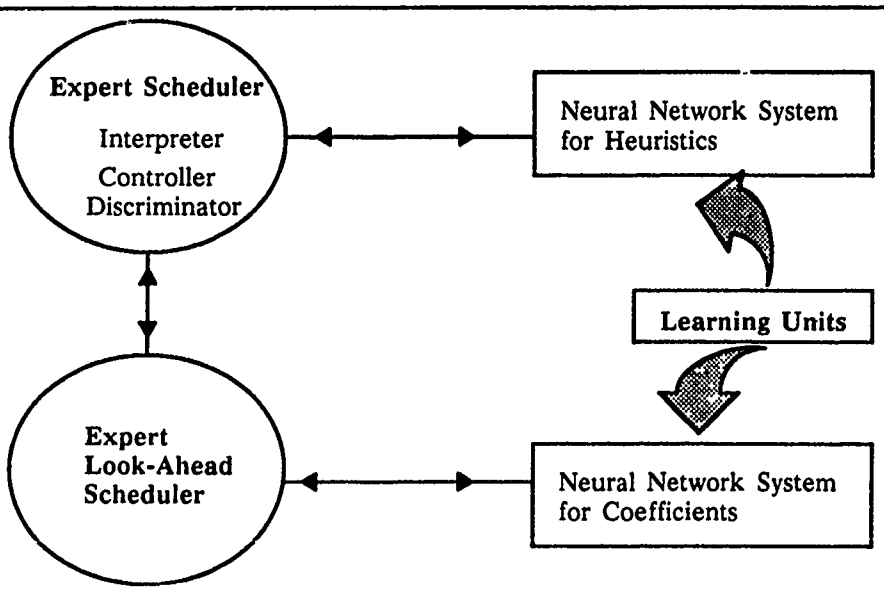


Figure 3.2 - Architecture of the Scheduler for the FMS

3.2.3 Flexible Manufacturing Systems - Rabelo, et al. (1990)

According to the authors' definition, flexible manufacturing systems (FMS) are automated manufacturing systems consisting of numerical control, machine tools, material handling devices, automated inspection stations, in-process storage areas, and a computational scheme to provide database handling, supervisory, and monitoring functions. A hybrid system that integrates neural networks and KBES is proposed to solve the real time scheduling of a flexible manufacturing system. Feedforward neural networks are used as prediction tools and scheduling pattern recognition mechanism, and KBES are utilized as the higher order members that interact with other elements of the FMS hierarchy providing guidance for problem solving strategy, monitoring the performance of the system, and automating the neural networks learning process. The architecture of the system are schematically shown in Fig. 3.2.

It appears that the architecture proposed in the article has direct applicability to scheduling and planning problems in construction engineering. The question is how effective the learning units will be on large complex data sets and how efficient the interfaces between expert system and neural networks are in real application.

3.2.4 Task of Ordering - Becker and Peng (1987)

This paper discusses the use of activation networks for analogical reasoning in the task of ordering the alternatives. The scheme for integrating the activation networks with a KBES for symbolic processing is also outlined. The activation network is designed to represent analogical reasoning for problem solving. There are three layers of nodes in the network and the characteristics of each layer are as follows: 1) the input layer represents problem attributes, 2) the hidden or middle layer represents old solutions, and 3) the output layer represents choice alternatives. Connections and connection strength between input node to hidden nodes are established if the input attribute contributes to the solutions in the hidden layer, and the connection strengths between the hidden layer to the output layer are assigned identity values. Learning schemes such as the parameter-adjusting learning can be used to

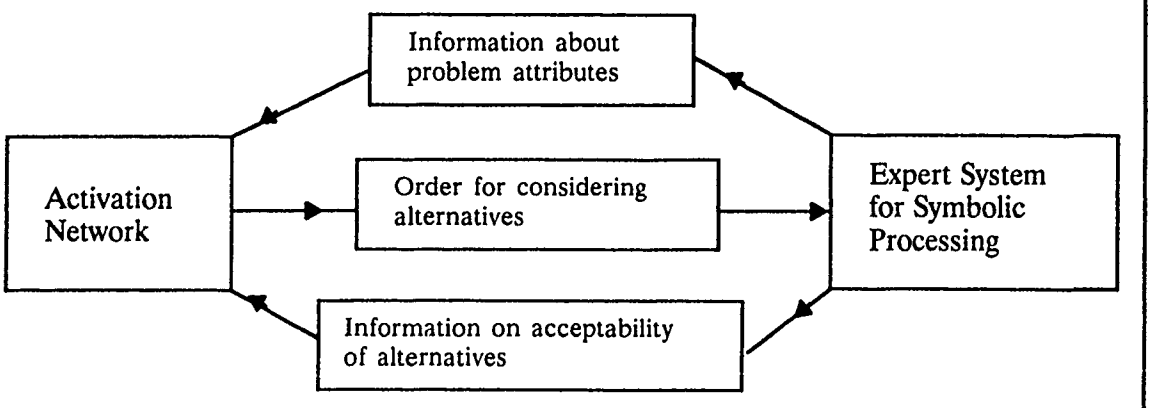


Figure 3.3 - The Integration of an Activation Network with KBES

adjust the weight between input and hidden layer. No learning example is shown in the paper. The integration of an activation network with an expert system can be schematically shown in Fig. 3.3.

3.2.5 Delivery Truck Dispatching – Bigus and Goosbey (1990)

This article describes the application of a self-organizing network with database and knowledge-based systems to solve the problem of dispatching delivery trucks under weight and volume constraints to minimize the number of trucks required and the total distance each truck must travel. The problem solving process involves four steps: 1) reading the data from a customer and delivery database and determining the minimum number of trucks required from KnowledgeTool – a rule-based expert system, 2) using KnowledgeTool rules for the initial assignment of deliveries to trucks, 3) using KnowledgeTool to improve the assignments by swapping deliveries between trucks to reduce the travelling distance, and 4) solving each truck's delivery route using a variation of elastic net proposed by Angeniol, et al. (1988) based on feature maps. The problem solving process is shown in Fig. 3.4.

3.2.6 Object Recognition in Image Processing – Glover, et al. (1990)

This article describes a hybrid system for object recognition in image processing. The system which is composed of neural networks and a rule-based pattern recognition system, is capable of self-modification or learning through a feedback loop between the neural networks and the rule-based system. Thus the neural networks can be automatically trained and modified by the rule-based system, and the rule-based system can modify models in this knowledge base from information supplied by the neural networks. The schematic diagram of the hybrid system architecture is shown in Fig. 3.5.

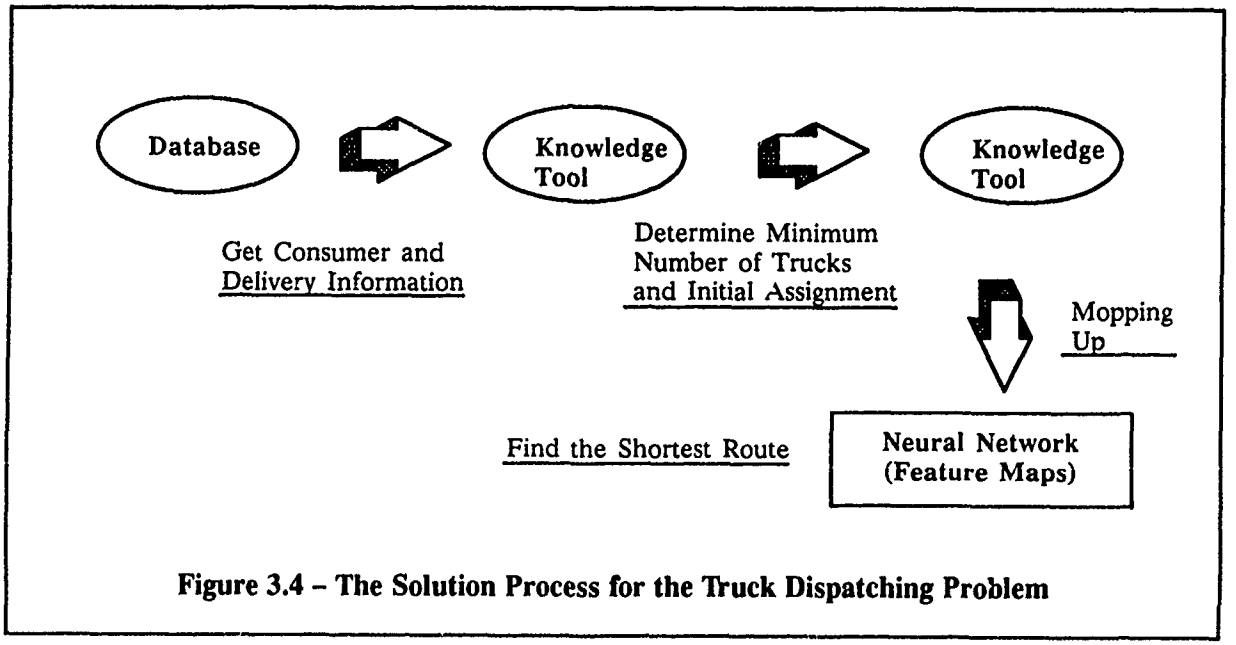


Figure 3.4 – The Solution Process for the Truck Dispatching Problem

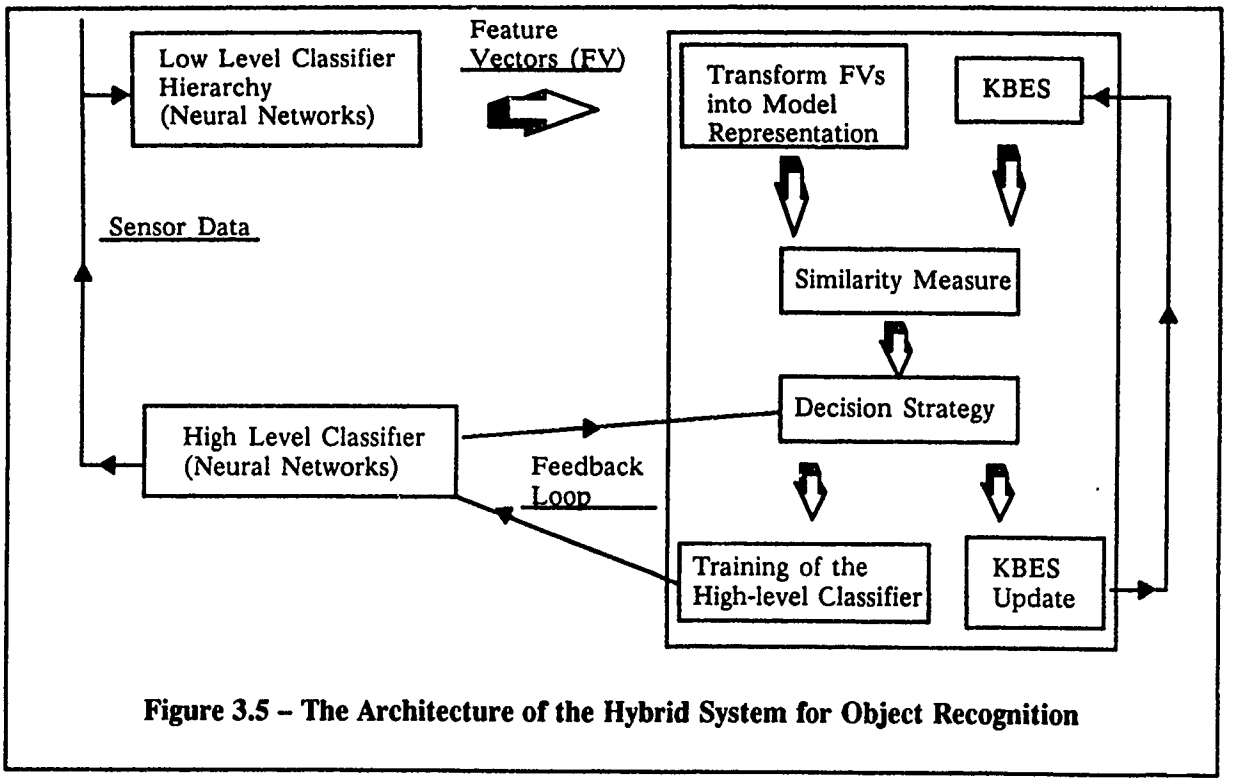


Figure 3.5 – The Architecture of the Hybrid System for Object Recognition

3.2.7 Manipulator Control System – Handelman, Lane and Gelgand (1990)

This article describes a hybrid system combining neural networks and rule-based systems for robot control problems. Like the system proposed by Glover, et al. (1990), the rule–based system interacts with and monitors the learning and performance of the neural network module. The training of the network can then be completed on line so an autonomous learning system is defined. The system works in the following way: 1) Initially, a rule-based system is used to come up with acceptable first-cut solutions to the given control objectives, 2) The rule-based system then teaches a neural network how to accomplish parts of the learning task, and 3) After that, the rule-based system interacts with, monitors the neural network operation, and re–engages task execution and training rules whenever changes in operating conditions degrade network performance. The schematic architecture of the control system for teaching a two-link manipulator to make a tennis-like swing is shown in Fig. 3.6.

3.2.8 Waste Water Treatment Sequence Processing – Krovvidy and Wee (1990)

This article describes a system for constructing waste water treatment sequences for the treatment of several compounds by reducing the concentration levels of the chemicals. The objective of the work is to extract the information from an existing database in the form of a collection of expert system rules and use these rules to come up with the treatment train. The system consists of two phases: the analysis phase and synthesis phase. The expert system rules obtained in the analysis phase are developed using an inductive algorithm and the treatment train is determined in the synthesis phase using a Hopfield network. The system is schematically shown in Fig. 3.7.

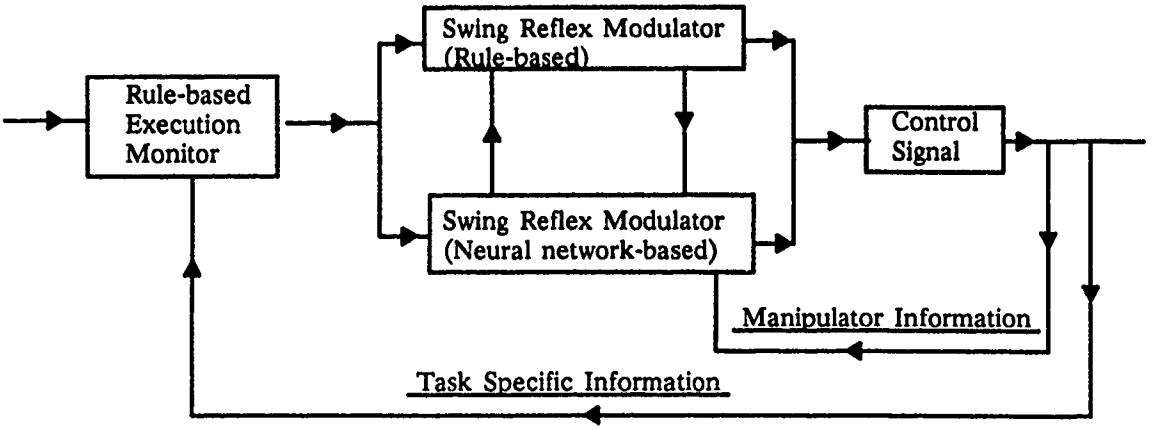


Figure 3.6 – Architecture of the Autonomous Learning System for Manipulator Control

3.2.9 Synthetic Organic Chemistry – Luce and Govind (1990)

This system can be called a hybrid system but not in the sense used here, because it is basically a compound system with several neural networks carrying out different operations. What we are really looking at the hybrid system for is the integration of neural networks with knowledge based or traditional systems instead of simply putting some neural networks (may be of different types) bundled

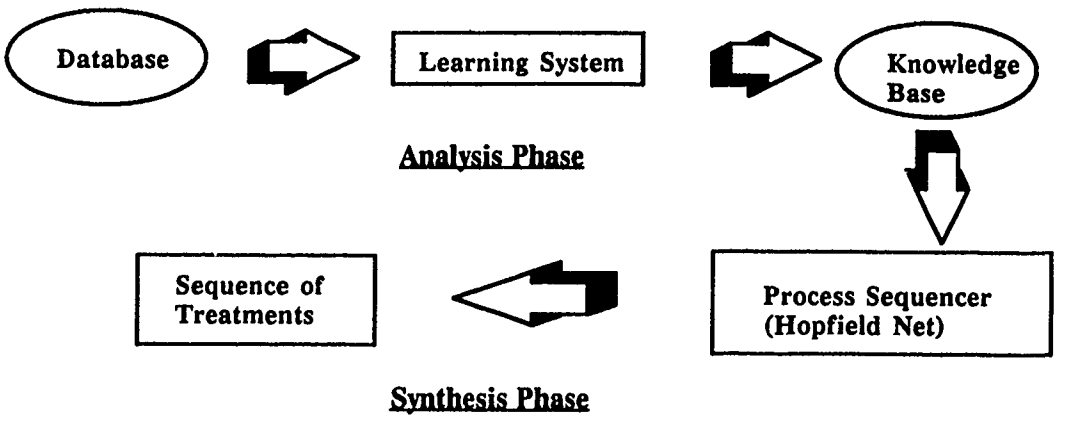


Figure 3.7 – Architecture of the Process Sequencer for Waste Water Treatment

together, though such bundled systems are also of significance in neural network research and applications.

In this paper, different neural networks are designed for the pattern recognition of molecular subgroups in organic molecules in producing and devising creative syntheses, and each network recognizes one set of disconnection type. The bundled system design was inspired by Minsky's "society of mind." The system design is shown in Fig. 3.8.

3.2.10 Fault Diagnosis – Yamamoto and Venkatasubramanian (1990)

This article proposed a novel approach to the fault diagnostics problems especially with multiple faults. The conventional neural network approach to fault diagnosis solely uses the feedforward mappings in an open loop, whereas the approach proposed here consists of two mapping operations, namely, feedforward mapping and inverse mapping. The inverse mapping networks give verification to the results, provide credibility to the output values of the forward mapping networks, and reduce the ambiguity in generalization. The system consists of three main components: quantitative neural networks (QTN), qualitative neural networks (QLN), and inverse qualitative neural networks (IQLN). Each module is also comprised of multiple networks with the same structure, and there are eight subnetworks in QTN, five in both QLN and IQLN, respectively. All three networks are feedforward backpropagation networks with one hidden layer. The general architecture of the system is shown in Fig. 3.9.

The performance of the system was evaluated through testing on the following four fault cases in a chemical plant model: 1) single-fault cases, 2) two-fault cases I, where one fault resides in the reactor and another in the distillation column, 3) two-fault cases II, where both faults are within the reactor, and 4) sensor fault cases. The performance of the system is reasonable in that it identified all the novel

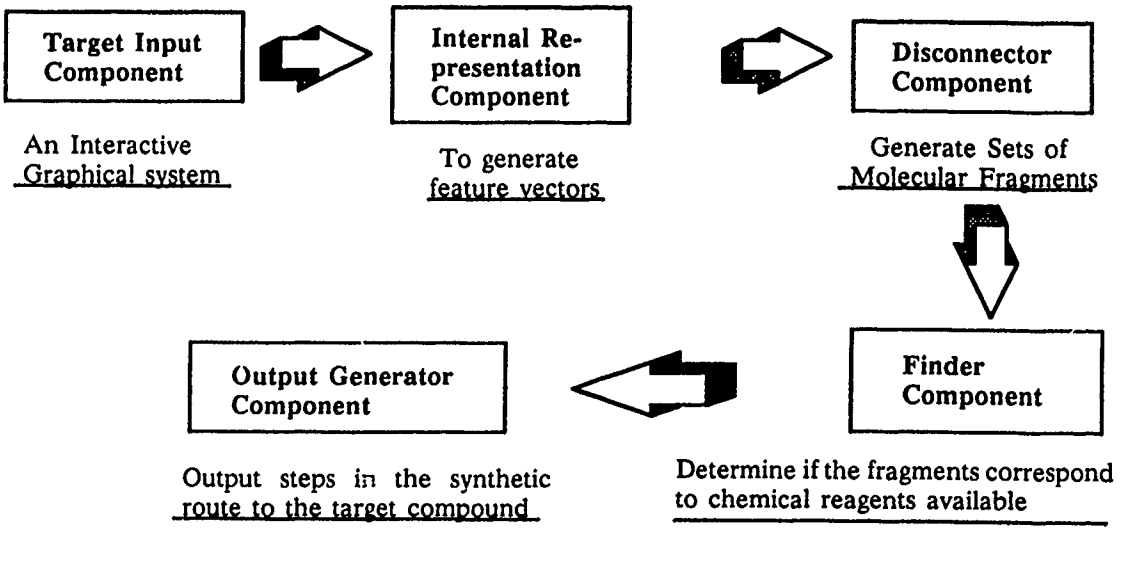


Figure 3.8 – System Design for Retrosynthetic Analysis in Synthetic Organic Chemistry

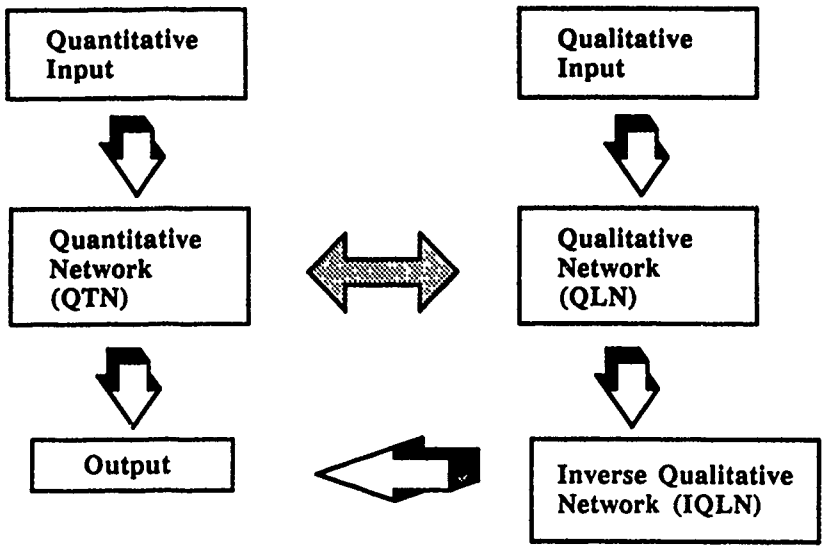


Figure 3.9 - System Design for Fault Diagnosis of a Chemical Plant Model

testing single-fault cases, two-fault cases I, and the sensor-fault cases. For two-fault cases II, half of the test patterns were correctly identified.

References

1. Angeniol, B., De La Croix Vaubois, G., and Le Texier, J.-Y., "Self-Organizing Feature Maps and the Traveling Salesman Problem," *Neural Network,* Vol. 1, pp. 289-293, 1988.

2. Becker, L. A., and Peng, J., "Using Activation Networks for Analogical Ordering of Consideration: One Method for Integrating Connectionist and Symbolic Processing," *Proceedings of the First IEEE International Conference on Neural Networks*, San Diego, 1987, II-367.

3. Benachenhou, D., Cader, M., Szu, H., Medsker, L., Wittwet, C., and Garling, D., "Neural Networks for Computing Invariant Clustering of a Large Open Set of DNA-PCR-Primers Generated by a Feature-Knowledge Based System," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 1990, II-83.

4. Bigus, J. P., and Goosbey, K., "Integrating Neural Networks and Knowledge-based Systems in A Commercial Environment," *Proceedings of the International Joint Conference on Neural Networks*, Washington, D. C., January 1990, II-463.

5. Carpenter, G. A., and Grossberg, S., "A Massively Parallel Architecture for a Self-organizing Neural Pattern Recognition Machine," *Computer Vision, Graphics, and Image Processing* 37, pp. 54-115, 1987.

6. Glover, C. W., Silliman, M., Walker, M., Spelt, P., and Rao, N. S., "Hybrid Neural Network and Rule-based Pattern Recognition System Capable of Self-modification," *SPIE Vol. 1293, Applications of Artificial Intelligence VIII*, M. M. Trivedi, Editor, pp. 290-301, 1990.

7. Handelman, D. A., Lane, S. H., and Gelgand, J.J., "Integration of Knowledge-Based System and Neural Network Techniques for Autonomous Learning Machines," *Proceedings of the International Joint Conference on Neural Networks*, Washington, D. C., June 18-22, 1989.

8. Hillman, D. V., "Integrating Neural Nets and Expert Systems," *AI Expert*, pp. 54-58, June 1990.

9. Krovvidy, S., and Wee, W. G., "A Knowledge Based Neural Network Approach for Waste Water Treatment System," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 1990, I-327.

10. Luce, H. H., and Govind, R., "Retro: An Expert System Which Embodies "Chemical Intuition," *Proceedings of the International Joint Conference on Neural Networks*, Washington, D. C., January 1990, II-643.

11. Luce, H. H., and Govind, R., "Neural Network Applications in Synthetic Organic Chemistry: I. A Hybrid System which Performs Retrosynthetic Analysis," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 1990, I-345.

12. Minsky, M., *The Society of Mind*, Simon & Schuster Inc., New York, 1985.

13. *NeuroShell*, Ward systems Group, Inc., 245 West Patrick St., Frederick, MD 21701.

14. Rabelo, L. C., and Alptekin, S., "Using Hybrid Neural Networks/Expert Systems for Intelligent Scheduling in Flexible Manufacturing Systems," *Proceedings of the International Joint Conference on Neural Networks*, Washington, D. C., pp. 608, June 18-22, 1989, Vol. II.

15. Rabelo, L. C., Kim, D., and Erdogen, T., "Integrating Digital and Artificial Neural Networks Using Neurocontrollers: an Intermediate Step Toward the Universal Computer," *Proceedings of the International Joint Conference on Neural Networks*, Washington, D. C., January 1990, II-191.

16. Rabelo, L. C., Slptekin, S., and Kiran, A. S., "Synergy of Artificial Neural Networks and Knowledge-Based Expert Systems for Intelligent FMS Scheduling," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 1990, I-359.

17. Yamamoto, Y., and Venkatasubramanian V., "Integrated Approach Using Neural Networks for Fault Detection and Diagnosis," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 1990, I-317.

3.2.11 Other Applications

There are many other applications that use a hybrid system in the problem solving process. Hybrid systems have been proposed and used for many suitable problems in practice, such as the monitoring and diagnostics, process control and optimization.

For diagnostic problems, Casselman and Acres (1990) use several neural networks in a large diagnostic system - the DASA/LARS, on monitoring and diagnosing spectrum anomalies associated with the Frequency Multiple Access satellite communication networks. The Neural Networks are trained

on live sensor information in an operational environment. For medical diagnostics, Saito and Nakano (1988) built a prototype medical diagnostic expert system based on a three layer backpropagation network trained on symptoms and diagnosis cases of about 300 patients. The system maps headaches as the only symptoms to 23 diseases. The input layer represents answers to each question of the 230 on a questionnaire. A scheme is also proposed to extract symbolic knowledge from the diagnostic results of the neural networks and compare with doctor's knowledge. In veterinary medicine, a system for diagnosis of mastitis in dairy cows is proposed by integrating a production system module, a neural network simulation module and a knowledge acquisition module (Schreinemakers and Touretzky, 1990). The production system is OPS5 and all three modules communicate via OPS5 working memory elements. The performance of the system on the diagnosis of mastitis in a limited set of experimental data showed excellent accuracy. In engineering facility management, monitoring and diagnostics are an integrated set. Tsoukalas and Reyes-Jimenes (1990) proposed a prototypic system for the monitoring and diagnosis of a nuclear plant model. A backpropagation neural network with one hidden layer is used to capture the correlation between sensor signals and the working state of different units. The rule-based system is used to interpret the results from the neural network and to make decisions and control the operations of working units.

Tsutsumi (1989) has done a series of work in robot and position control. Though his system does not fit in the pure mode of hybrid systems, his approach is worth mentioning. The article describes a system consisting of two backpropagation networks and a Hopfield network for applications in manipulator control. The input signals from the environment are mapped via a backpropagation network into the internal space, where a Hopfield net minimizes the total energy according to the internal space representation. The output signals of the Hopfield Net are then mapped into the environment via another backpropagation net with the inverse mapping. Simulation studies on manipulator configuration control how the proposed system helps the manipulator to reach the target point through the shortest path in the internal space. It should be pointed out that a similar approach on adding an inverse mapping in the control loop was also proposed in Yamamoto and Venkatasubramanian's study (1990) on a chemical plant control model.

Though the neural network approach to planning is discussed in another section in this chapter, Veezhinathan and McCormick's work on plan reminding (1988) is interesting. As defined in the article, plan reminding is the process by which we are reminded of a plan or a set of plans to achieve a given goal or a combination of goals by taking into account certain familiar constraints automatically. The characteristics of plan reminding is as follows: 1) it is indexed not only by goals, but also depends critically on the context in which the goals occur, 2) constraint satisfaction is an important consideration, and 3) it may involve inference. This article describes a prototypic connectionist model for the task of errand planing in plan reminding within a production system.

References

1. Casselman, F., and Acres, D. J., "DASA/LARS: A Large Diagnostic System Using Neural Networks," *Proceedings of the International Joint Conference on Neural Networks*, Washington, D. C., January 1990, II-539.

2. Fu, Li-Min, "Integration of Neural Heuristics Into Knowledge-Based Inference," *Proceedings of the International Joint Conference on Neural Networks*, Washington, D. C., pp. 606, June 18-22, 1989, Vol. II.

3. Kudrychi, T. P., *Neural Network Implementation of a Medical Diagnosis Expert System*, M. S. Thesis, Dept. of Electrical and Computer Engineering, University of Cincinnati, 1988.

4. Martinez, O. E., and Harston, C., "Interfacing Data Base to Find the Best and Alternative Solutions to Problems by Obtaining the Knowledge from the Data Base," *Proceedings of the International Joint Conference on Neural Networks*, Washington, D. C., January 1990, II-663.

5. McMahon, D. C., "A Neural Network Trained to Select Aircraft Maneuvers During Air Combat: A Comparison of Network and Rule Based Performance," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 1990, I-107.

6. Saito, Kazumi, and Nakano, Ryohei, "Medical Diagnostic Expert System Based On PDP Model," *Proceedings of the Second IEEE International Conference on Neural Networks*, San Diego, pp. 252-262, 1988, Vol. I.

7. Schreinemakers, J. F., and Touretzky, D. S., "Interfacing a Neural Network with a Rule-based Reasoner for Diagnosing Mastitis," *Proceedings of the International Joint Conference on Neural Networks*, Washington, D. C., January 1990, II-487.

8. Selinsky, J., Guez, A., Eilbert, J., and Kam, M., "A Neuro-Expert Architecture for Object Recognition," *Proceedings of the International Joint Conference on Neural Networks*, Washington, D. C., June 18-22, 1989, Vol. II, pp. 574.

9. Sharda, R., and Patil, R. B., "Neural Networks as Forecasting Experts: An Empirical Test," *Proceedings of the International Joint Conference on Neural Networks*, Washington, D. C., January 1990, II-491.

10. Tsoukalas, L., and Reyes-Jimenez, J., "Hybrid Expert System: Neural-network Methodology for Nuclear Plant Monitoring and Diagnostics," *SPIE Vol. 1293, Applications of Artificial Intelligence VIII*, M. M. Trivedi, Editor, pp. 1024-1031, 1990.

11. Tsutsumi, K., "A Multi-layered Neural Network Composed of Backprop, and Hopfield Nets and Internal Space Representation," *Proceedings of the International Joint Conference on Neural Networks*, Washington, D. C., 1989, I-365.

12. Ursic, S., "Synthesis of Neural Networks with Linear Programs," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 1990, I-379.

13. Veezhinathan, J., and McCormick, B. H., "Connectionist Plan Reminding in a Hybrid Planning Model," *Proceedings of 1988 IEEE International Conference on Neural Networks*, II-515.

14. Voevodsky, John, "PLATO/ARISTOTLE: A Neural Network Knowledge Processor," *Proceedings of the First IEEE International Conference on Neural Networks*, San Diego, 399-407, 1987, Vol. II.

15. Yager, R. R., "On the Aggregation of Processing Units in Neural Networks," *Proceedings of the First IEEE International Joint Conference on Neural Networks*, San Diego, 1987, II-327.

3.2.12 Studies on Connectionist Expert Systems and Related Works

Some works concerning the methodology in building connectionist expert systems and schemes for rule extraction, concept mapping, generalization, etc., are also listed in the following references.

For example, Bochereau and Bourgine (1990) discussed the extraction of logical rules from a multilayer neural network through building a validity domain for the network. The work is of significance in theoretical study in building real neural expert systems with explicit knowledge about the internal reasoning. However, its applicability is still an open question as no case studies were given in the article. In Miller, Roysam and Smith's work (1988), a general method for mapping a large class of rule-based constraints to their equivalent stochastic Gibbs' distribution representation was proposed. This mapping thus makes it feasible to solve stochastic estimation problems over rule-generated constraint space within a Bayesian framework. The algorithm was also tested on a image reconstruction problem.

In addition to Gallant's (1988) seminal work on the construction of connectionist expert systems, new schemes have also been proposed in that respect, such as the work by Yang and Bhargava (1990), Touretzky, et al. (1986, 1987, 1990), Samad (1988) and Fu (1989). Some works also cover the integration of fuzzy logic with neural networks (Kosko, 1987; Romaniuk and Hall, 1990).

References

1. Bochereau, L., and Bourgine, P., "Extraction of Semantic Features and Logical Rules from a Multilayer Neural Network," *Proceedings of the International Joint Conference on Neural Networks*, Washington, D. C., January 1990, II-579.

2. Bochereau, L., and Bourgine, P., "Rule Extraction and Validity Domain on A Multilayer Neural Network," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 1990, .

3. Dolan, C. P., and Dyer, M. G., "Symbolic Schemata, Role Binding, and the Evolution of Structure in Connectionist Memories," *Proceedings of the First IEEE International Joint Conference on Neural Networks*, San Diego, 1987, II-287.

4. Drucker, H., "Implementation of Minimum Error Expert System," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 1990, I-291.

5. Fozzard, R., Bradshaw, G., and Ceci, L., "A Connectionist Expert System that Actually Works," in *Advances in Neural Information Processing Systems I*, D. S. Touretzky, Editor, Morgan Kaufmann Publishers, Inc., pp. 248, 1989.

6. Fu, Li-Min, "Building Expert Systems on Neural Architecture," *The First IEE International Conference on Artificial Neural Networks 1989*, The University of Wisconsin-Milwaukee, pp. 221-225, October 1989.

7. Gallant, S. I., "Automatic Generation of Expert Systems from Examples," *Proceedings of the International Conference on Artificial Intelligence Applications II*, 1985.

8. Gallant, Stephen I., "Connectionist Expert Systems," *Communications of the ACM*, 31(2), pp. 152-169, Feb. 1988.

9. Goodman, R. M., Miller, J. W., and Smyth, P., "An Information Theoretic Approach to Rule Based Connectionist Expert Systems," in *Advances in Neural Information Processing Systems I*, D. S. Touretzky, Editor, Morgan Kaufmann Publishers, Inc., pp. 256, 1989.

10. Hsu, L. S., Teh, H. H., Chan, S. C., and Loe, K. F., "Fuzzy Logic in Connectionists' Expert Systems," *Proceedings of the International Joint Conference on Neural Networks*, Washington, D. C., January 1990, II-599.

11. Huffman, J., and Scoggins, J., "Architectural Isomorphisms in Neural Network Applications," *Proceedings of the International Joint Conference on Neural Networks*, Washington, D. C., January 1990, II-603.

12. Hutchison, W. R., and Stephens, K. R., "Integration of Distributed and Symbolic Knowledge Representations," *Proceedings of the First IEEE International Conference on Neural Networks*, San Diego, 1987, II-395.

13. Jacobs, R. A., "Initial Experiments on Constructing Domains of Expertise and Hierarchies in Connectionist Systems," *Proceedings of the 1988 Connectionist Models, Summer School*, D. Touretzky, G. Hinton and T. Sejnowski (Eds.), pp. 144-153, Carnegie Mellon University, Morgan Kaufman Publishers, Inc., 1989.

14. Kadaba, N., Nygard, K. E., Juell, P. L., and Kangas, L., "Modular Backpropagation Neural Networks for Large Domain Pattern Classification," *Proceedings of the International Joint Conference on Neural Networks*, Washington, D. C., January 1990, II-551.

15. Kaplan, D., and Johnson, D. L., "HNETTER - A Heuristically Driven Neural Reasoning System," *Proceedings of 1988 IEEE International Conference on Neural Networks*, II-507.

16. Kosko, Bart, "Adaptive Inference in Fuzzy Knowledge Networks," *Proceedings of the First IEEE International Conference on Neural Networks*, San Diego, Vol. II, pp. 261-268, 1987.

17. Miller, M. I., Roysam, B., and Smith, K. R., "Mapping Rule Based and Stochastic Constraints to Connection Architectures: Implication for Hierarchical Image Processing," *SPIE*, 1078-1085, 1988.

18. Myllymaki, P. Tirri, H., Floreen, P., and Orponen, P., "Compiling High-Level Specifications into Neural Network," *Proceedings of the International Joint Conference on Neural Networks*, Washington, D. C., II-475, January 1990.

19. Orponen, P., Floreen, P., Myllymaki, P., and Tirri, H., "A Neural Implementation of Conceptual Hierarchies with Bayesian Reasoning," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 1990, I-297.

20. Pham, K. M., and Degoulet, P., "MOSAIC: A Macro-Connectionist Organization System for Artificial Intelligence Computation," *Proceedings of 1988 IEEE International Conference on Neural Networks*, II-533.

21. Romaniuk, S. G., and Hall, L. O., "Fuzznet: Towards a Fuzzy Connectionist Expert System Development Tool," *Proceedings of the International Joint Conference on Neural Networks*, Washington, D. C., January 1990, II-483.

22. Samad, Tariq, "Toward Connectionist Rule-based systems," *Proceedings of the Second IEEE International Conference on Neural Networks*, San Diego, Vol. II, pp. 525–532, 1988.

23. Shastri, Lokendra, and Ajjanagadde, Venkat, "From Simple Associations to Systematic Reasoning: A Connectionist Representation of Rules, Variables and Dynamic Bindings," *MS–CIS–90–05*, Department of Computer and Information Science, University of Pennsylvania, PA, January 1990.

24. Teh, H. H., and Tan, A. H., "Connectionist Expert Systems – A Neural-Logic Models' Approach," *Internal Faculty Seminar on Neural Net Computing DISCS NUS*, pp. 16–32, June 1989, Institute of Systems Science, National University of Singapore.

25. Touretzky, David S., and Hinton, Geoffrey E., "A Distributed Connectionist Production System," *CMU–CS–86–172*, Department of Computer Science, Carnegie–Mellon University, Pittsburgh, PA.

26. Touretzky, David S., "Representing Conceptual Structures in a Neural Network," *Proceedings of the First IEEE International Joint Conference on Neural Networks*, San Diego, 1987, II–279.

27. Touretzky, D. S., and Elvgren III, G., "Rule Representations in a Connectionist Chunker," *Advances in Neural Information Processing Systems 2*, pp. 431–438, Morgan Kaufmann Publishers San Mateo, CA, 1990.

28. Yang, Q., and Bhargava, V. K., "Building Expert Systems by a Modified Perceptron Network with Rule-Transfer Algorithms," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 1990, II–77.

3.3 Civil Engineering Applications

The application of neural networks in civil engineering ranges from modeling of material behaviors from experimental tests, damage assessment of structural systems, structural analysis and optimum design, to ground water modeling. Research works in this area are still exploratory and the question is on: 1) the definition of the specific application domains, namely, in what subdiscipline or subarea neural networks would most probably provide advantage and benefits over the current approach? 2) what kind of network is most suitable for problems in civil engineering? The examples shown in the following paragraphs will provide some test cases for judging the pros and cons for neural network approaches in this field.

3.3.1 Modeling of the Behaviors of Engineering Materials

The application of neural networks to material modeling was initiated in the Department of Civil Engineering at the University of Illinois. The focus of this research group is on the modeling of behaviors of engineering materials such as concrete, reinforced concrete, geo-materials, as well as composites, the assessment of structural member damage in a structural system, and the classification of groundwater transmissivity fields for use in the design of groundwater contamination remediation. In material modeling, the behavior of a material under different stress states determined from experiments is represented as a kind of knowledge in a neural network.

The paper published in the Proceedings of NUMETA-90 (Ghaboussi, et al. 1990) described a neural network approach to the modeling of engineering material, specifically plain concrete. The complex behavior of concrete material in biaxial stress states under tension–tension, tension-compression and compression–compression monotonic loading was modeled by a backpropagation neural networks with two hidden layers. The neural network–based model is stress controlled, that is, it predicts strain increments from information on the current stress–strain states and stress increments on a stress path. The neural network concrete model learned the behavior with reasonable accuracy and its predictions on untrained stress paths were on par with those predicted from analytical models.

The article in the ASCE journal (Ghaboussi, et al. 1991) formally proposed a neural network–based material modeling methodology for engineering materials with complex mechanical behavior. Behaviors of plain concrete in biaxial stress states under monotonic loading and those in uniaxial stress state under cyclic compressive loading were modeled in backpropagation neural networks. A "3–point scheme" was used to represent the history dependency of material behavior under cyclic loading. Comprehensive testing has been carried out to verify the neural network–based models with additional experiments and analytic models based on principles of solid mechanics. Implication of the network–based modeling methodology to the difficult problem of composite material modeling was also outlined.

References

1. Garrett, Jr., J. H., Ghaboussi, J., and Wu, X., "Neural Networks," a chapter in *Expert Systems in Civil Engineering – Knowledge Representation*, R. H. Allen, Ed., The Expert System Committee of the American Society of Civil Engineers, 1991.

2. Ghaboussi, J., Garrett, Jr., J. H., and Wu, X., "Material Modelling with Neural Networks," *NUMETA-90: Proceedings of the International Conference on Numerical Methods in Engineering: Theory and Applications*, Swansea, U. K., January 1990, pp. 701–717.

3. Ghaboussi, J., Garrett, Jr., J. H., and Wu, X., "Knowledge-Based Approach to Material Modelling with Neural Networks," in *J. of Engineering Mechanics*, American Society of Civil Engineers (ASCE), January 1991.

4. Wu, X., Garrett, Jr., J. H., and Ghaboussi, J., "Representation of Material Behavior: Neural Network-Based Models," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17–21, 1990.

3.3.2 Structural Analysis and Design

3.3.2.1 Deb's Approach (1990)

The design of a welded beam involves the determination of a feasible set of geometrical parameters for the weld and the beam under a concentrated load at the free end, which the system subjects to constraints on the shear stress in the weld, bending stress in the beam, buckling loading, beam end displacement, and the width of the weld. Genetic Algorithms (GAs) are used to solve the problem with

four design parameters and five sets of constraints. Performances of GAs with three population sizes (100, 50, 200) and corresponding probabilities of crossover and mutation are compared with traditional optimization method. The final results are very reasonable and the maximum error for the population of 100 case is about 3 percent.

The parameters used in the GA are: population size (100, 50, 200), string length (40), sub-string length for each parameter (10), probability of cross over (0.9, 0.5–1.0), and probability of mutation (reciprocal of the population size). Because a GA only needs to search for an optimal solution in a subspace of the feasible space, GAs look very promising in structural optimization problems. On the other hand, a genetic search procedure is implicitly parallel and thus provides fast searching capabilities especially on parallel machines.

3.3.2.2 Hajela and Berke's Approach (1990)

In structural design, major computation is on the analysis of the structural behavior with different sets of design parameters under designated loading. In this paper, neural network models are used to replace the structural analysis module in a nonlinear-programming-based optimization environment. The feedforward backpropagation network and the functional-link net were used to capture the load-displacement relationships in static structural analysis in the minimum weight design of a five-bar truss, a ten-bar truss, a wing-box structure, with constant nodal loading and constraints on maximum nodal displacements or axial stresses or both.

From network performance study, it was found that the functional-link net was not very effective in highly nonlinear mapping of the load-displacement relation. The identification of proper input enhancement would be the key for the success of functional-link net, and this task is not easy to achieve. There were 16 design variables and 40 design constraints on both displacements and stresses for the wing-box structure, and the training sets covered possible lower and upper bounds of each design variable. For each design case, each input node represents a design variable (cross-sectional area) and each output node represents a nodal displacement. It appeared that the neural networks achieved the near-optimum design for each structure.

In our opinion, the advantage of this approach is the fast mapping of load-displacement relation after training. The pitfall lies in the following aspects: 1) the structural geometry is fixed, i.e., there needs be a neural networks for each structural configuration, and 2) a large number of training sets are needed to cover the lower and upper bounds for each design variables. For large real structures, it would be too expensive to generate all the training data. A comparison with genetic algorithms may shed some light on this problem because only a small subspace of the training domain is needed with genetic searching algorithms (Goldberg, 1989).

3.3.2.3 Mcaulay's Approach (1987)

This paper also addressed the application of a backpropagation network in structural design and a new learning algorithm called "split inversion learning." The basic approach of this algorithm is to compute the weights for the output and hidden layers separately so that the error at the output layer is minimized. In structural design, an inverse problem to structural analysis is solved by a backpropaga-

tion network to model the displacement-load relation under varying magnitudes of loadings. For a static linear beam-truss problem, the loads and controlling displacements are represented as input information, and the design variables such as the dimension of each structural member as output information. No numerical solution was shown in the paper.

Further study needs to be done on the proposed new learning algorithm before it becomes a viable one. As discussed, only if the number of data sets is equal to the dimension of the output layer, can the update of weights be obtained from solving a linear system of equations. In this case, two systems of linear equations corresponding to that from input layer to output layer and that from hidden layer to output, need to be solved. If the condition is not satisfied, or if the number of training sets is larger than the dimension of the output layer, which is the usual case in practice, then a linear squares problem must be solved instead. This made the learning algorithm awkward and inefficient. On the other hand, the remarks suggesting the conjugate gradient method for the ill-conditioned linear system of equations were misleading because conjugate gradient even with simple preconditioner is also ineffective for those highly ill-conditioned systems.

3.3.2.4 Rehak, Thewalt, and Doo's Approach (1989)

This paper is probably the first one addressing the application of neural networks to structural mechanics in civil engineering. By considering its summation rules for a neuron in a neural network, a neural model of spring structure is construed as computational elements in structural mechanics, and its use in system identification computation for a dynamic system, specifically a one degree of freedom oscillator with viscous damping, was illustrated. Since this approach was basically drawn from the analogy of mapping properties associated with linear neural networks and the solution procedure in linear systems, the applicability of the approach is limited and does not offer any improvement on the current approach to structural system analysis. In spite of this, the major contribution of this paper lies in its realization of possible impacts of neural networks to system identification computation, which is also of significance in active structural control and structural dynamics.

From critically reviewing this article, one should recognize the limited scope of application of neural networks, that is, neural networks should be used in those areas where they have the potential of resulting in better performance and improvement over the currently used methodologies. This aspect needs innovative thinking and critical reasoning.

References

1. Deb, K., "Optimal Design of A Class of Welded Structures Via Genetic Algorithms," *AIAA-90-1179-CP*, American Institute of Aeronautics and Astronautics, pp. 444-453, 1990.
2. Deb, K., and Goldberg, D. E., "Natural Frequency Calculation Using Genetic Algorithms," *Proceedings of SECTAM XV*, Southeastern Conference on Theoretical and Applied Mechanics, 1990.
3. Hajela, P., and Berke, L., "Neurobioligical Computational Models in Structural Analysis and Design," *AIAA-90-1133-CP*, American Institute of Aeronautics and Astronautics, pp. 345-355, 1990.

4. Goldberg, D. E., *Computer-aided Gas Pipeline Operation Using Genetic Algorithms and Rule Learning*, Ph.D. dissertation, University of Michigan, 1983.

5. Goldberg, D. E., and Samtani, M. P., "Engineering Optimization via Genetic Algorithms," *Proceedings of the Ninth Conference on Electronic Computation*, pp. 471–482, 1987.

6. Goldberg, D. E., and Richardson, J., "Genetic Algorithms with Sharing for Multimodal Function Optimization," in *Genetic Algorithms and Their Applications: Proceedings of the Second International Conference on Genetic Algorithms*, pp. 41–49, 1987.

7. Goldberg, D. E., *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley Publishing Company, Inc., New York, 1989.

8. Mcaulay, A. D., "Engineering Design Neural Networks Using Split Inversion Learning," *Proceedings of the First IEEE International Conference on Neural Networks*, San Diego, 1987, IV–635.

9. Rehak, D. R., Thewalt, C. R., and Doo, L. P., "Neural Network Approaches in Structural Mechanics Computations," *Proceedings of the Sessions Related to Computer Utilization at Structural Congress'89*, American Society of Civil Engineers, pp. 168-176, May 1989.

3.3.3 Structural Damage Assessment and Fatigue Prediction

3.3.3.1 Garrett, et al. (1990)

The paper by Garrett, et al., is essentially a conglomerate of term project reports produced from Professor Garrett's class in the Department of Civil Engineering at the University of Illinois. It addresses the following prototypic engineering applications of neural computing: 1) an adaptive controller for building thermal mass storage, 2) an adaptive controller for adjustment of a combine harvester, 3) an interpretation system for non-destructive testing data on masonry walls for damage detection, 4) a machining feature recognition system in process planning, 5) an image processor for classifying land features from satellite images, and 6) a system for designing pumping strategy for contaminated groundwater remediation. Backpropagation networks were used in the first five applications and the Hopfield network was used for the last application. All the results reported in this paper are exploratory and preliminary, and further success rests on intensive work in those directions.

3.3.3.2 Troudet and Merrill (1990)

This paper described a neural network approach to estimate in real-time the fatigue life of mechanical components in the Intelligent Control Systems (ICS) for Reusable Rocket Engines (RRE) at the NASA Lewis Research Center. This fatigue life estimator consists of two functional blocks: a preprocessor and a neural network. The function of the preprocessor is to identify a load cycle and store the extreme load values in a shift-register buffer, which is then directly mapped to the input layer of a backpropagation neural network. The identification of a load cycle is based on the Uniaxial Local Strain Approach (Dowling, 1972; Palmgren, 1945; and Miner, 1945).

The architecture of the backpropagation network consists of 15 nodes in the input layer, 100 and 50 nodes in the first and second hidden layers respectively, and one node in the output layer. The input

nodes represent the content of the shift-register consisting of the load cycle boundaries and the peak-to-peak transitions of the previous cycle-free load history, and the output represents the contribution to the fatigue life of a hysteresis loop. The performance of the network based fatigue estimator is reasonable with 75 percent of the estimated values within a factor 1.7 of the exact data. No figures in the article showed the training and testing results.

3.3.3.3 Wu, Ghaboussi, and Garrett (1991)

A neural network approach has been proposed to the assessment of structural elements damage in a structural system from classifying deviations in the system behavior. From structural mechanics, it is realized that the response spectrum of a damaged structure in the frequency domain would differ by certain amount from that of the intact structure. Therefore, the damage of a structural system can be identified by a neural network if the network is trained on the response spectra corresponding to different damage states. A three story shear building was used in this study.

The approach has three computational steps: 1) the response of the structure under seismic excitation was determined in the time domain, 2) this response is then transformed into a response spectrum in the frequency domain through Fast Fourier Transformation (FFT), and 3) the normalized spectrum is then used for training a backpropagation network. Different architectures with one or two hidden layer(s) have been investigated, and it was found that the architecture had little effect on the performance of the trained network. The training results were perfect, and the test results on untrained cases were reasonable. Extended work is on the damage assessment of framed structural systems such as an offshore oil tower.

References

1. Garrett, Jr., J. H., Case, M. P., Hall, J. W., Yerramareddy, S., Herman, A., Sun, R., Ranjithan, S. and Westervelt, J., "Engineering Applications of Neural Networks," manuscript submitted for publication in *Journal of Intelligent Manufacturing*, 1990.
2. Miner, M. A., "Cumulative Damage in Fatigue," *J. of Applied Mechanics, Trans. ASME*, Vol. 12, pp. 159–164, 1945.
3. Norton, M. P., *Fundamentals of Noise and Vibration Analysis for Engineers*, Cambridge University Press, Cambridge, U. K., 1989.
4. Palmgren, A., *Ball and Roller Bearing Engineering*, SKF Industries, Inc., pp. 82–83, 1945.
5. Troudet, T., and Merrill, W., "A Real Time Neural Net Estimator of Fatigue Life," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17–21, 1990.
6. Wu, X., Ghaboussi, J., and Garrett, Jr., J. H., "Use of Neural Networks in Detection of Structural Damage," accepted for publication in *Computers and Structures*, 1991.

3.3.4 Groundwater Remediation

Several applications of neural networks in groundwater remediation are currently being investigated at the Department of Civil Engineering at the University of Illinois, using multilayer feedforward

and Hopfield type networks. They considered a hydraulic gradient control technique to determine the optimal pumping strategy for groundwater remediation under conditions of uncertainty. Their design method adopted a stochastic approach, where many equally likely sets of the uncertain parameter are included simultaneously in the design model. The major source of uncertainty is assumed to be embedded in the heterogeneity of the hydraulic conductivity parameter. Although many parameter fields are considered, only a few critical fields will impact the final design. The spatial distribution of the hydraulic conductivity values in a hydraulic conductivity field determines how critical that field is. The first application of a neural network was to train a feed-forward type network to learn, via error back-propagation, the association between a hydraulic conductivity field and its impact on the design. This network then classifies a large set of hydraulic conductivity fields according to their level of criticalness. Promising results have been obtained in this area of application. The trained neural network will be used as a prescreening tool, looking for the critical hydraulic conductivity fields, in the groundwater remediation design procedure. The pumping strategy for hydraulic gradient control is determined by solving an optimization model. The second application of the neural network was to set up a Hopfield style network to solve the optimization model through simultaneous constraint satisfaction. This approach enables the solution to optimization models with three components of pumping cost: cost of installation, cost of pump machinery, and cost of pump operation. Traditional linear programming techniques could run into computational complexities in this case. The preliminary results show that the neural network approach to optimization has the capability to converge to solutions that are optimal or near-optimal.

References

1. Garrett, Jr., J. H., Ranjithan, S., and Eheart, J. W., "Application of Neural Network to Groundwater Remediation," a chapter in *Expert Systems in Civil Engineering – Knowledge Representation*, R. H. Allen, Ed., The Expert System Committee of the American Society of Civil Engineers, 1991.

2. Ranjithan, S., Eheart, J. W., and Garrett, Jr., J. H., "Decision Making Under Uncertainty for Groundwater Reclamation: A Neural Network Approach," paper presented at the *American Geophysical Union Fall Meeting*, San Francisco, December 1989.

3. Ranjithan, S., Garrett, Jr., J. H., and Eheart, J. W., "A PDP Approach to Optimization: an Application to Hydraulic Gradient Control Model with Fixed Cost Components," *manuscript.*

3.4 System Identification

In engineering applications, a vital part of analysis is to model and estimate the behavior of a system from observations on input-output information. There are usually two steps in the estimation of a system behavior: functional estimation and parameter estimation. In a conventional sense, system identification is concerned with the determination of the parameters of a system after assuming a system function with unknown parameters is known. According to Zadeh (1962), the system identification problems can be defined as *"the determination, on the basis of input and output, of a system with a specified class of systems, to which the systems under test is equivalent."*

There are many ways to represent a system identification problem. In electrical control engineering, the early approach to system identification was concerned with the determination of the transfer function of a system. The transfer function could be determined by applying a known input signal to an assumed system, measuring the response of the system, and fine tuning the system parameters until the expected response was produced. This kind of approach is unsuitable for nonlinear systems or systems with measurement and process noises. Though the system parameters for a linear system can be determined in one step by solving a least squares problem, the determination of system parameters for a nonlinear system is an iterative process. A general form to represent both linear and nonlinear systems is the Kolmogorov-Garbor polynomial (Garbor, et al., 1961).

Neural networks with supervised learning, such as backpropagation networks, have been investigated for solving system identification problems because of the functional mapping capability of feedforward networks (Hornik, et al., 1989). It has been observed through experiments with the modeling and prediction of chaotic time varying systems that a feedback mechanism is necessary to identify a system with dynamic response. Details of this argument are illustrated in research described in this section. In other respects, the unique feature of neural network--based system identification methodology should be reiterated. Instead of obtaining explicit expressions about the system functions and corresponding parameters, a neural network solves a system identification problem through training on the input-output data sets observed. The underlying function and parameters of a system identified by a neural network are embodied in the network in a pattern of connection weights after proper training. In the following paragraphs, some of the typical approaches to system identification using neural networks are described.

3.4.1 Lapedes and Farber (1987)

This paper has probably become the classic in the neural network approach to prediction and system modeling. A standard three layer backpropagation neural network with one hidden layer is used to predict points in a highly chaotic time series by using a time window representation scheme. Three previous points on the time coordinate are presented to the input layer and response on the next time station is used as output in the output layer for prediction. For this problem, performance of the neural network is better than some conventional methods (such as the linear predictive method) in that the former gives orders of magnitude an increase in accuracy. An interesting experiment was also performed to study the underlining approximation capability of the network by using trigonometric sinusoid instead of the usual sigmoid function for the transfer function so that a generalized Fourier approximation resulted from the network.

3.4.2 Tenorio and Lee (1989)

Tenorio and Lee designed their Self-Organizing Neural Network (SONN) algorithm specifically for system identification problems by approximating function and functional parameters estimation in a unified process. The algorithm performs a search on the model space by the construction of hypersurfaces so the identification of a nonlinear system is viewed as the construction of an N + 1 dimension hypersurface when the system is represented by a Kolmogorov-Garbor polynomial with the highest order of N. The architecture of the network evolves while training takes place. The SONN consists of three processes: 1) a generating rule of the primitive neuron transfer functions, 2) an evaluation method which measures the quality of the model, and 3) a structure search strategy for adjusting the architecture of the network. When tested on modeling, the chaotic time series generated from the Mackey-Glass differential equations, SONN gives a satisfactory performance compared with Lapedes and Farber's work (1987).

3.4.3 Fernandez, Parlos and Tsai (1990)

This article describes a recurrent multi-layer perceptron network and its use in the identification of nonlinear dynamic systems based on input-output measurements. Feedback connections between layers and intralayer recurrent connections are introduced in the network, and the learning algorithm is a modified version of backpropagation learning rule. Using the network based system for the identification of a boiler model was satisfactory, yet perceivable training errors existed. The architecture of the neural network is shown in Fig. 3.10.

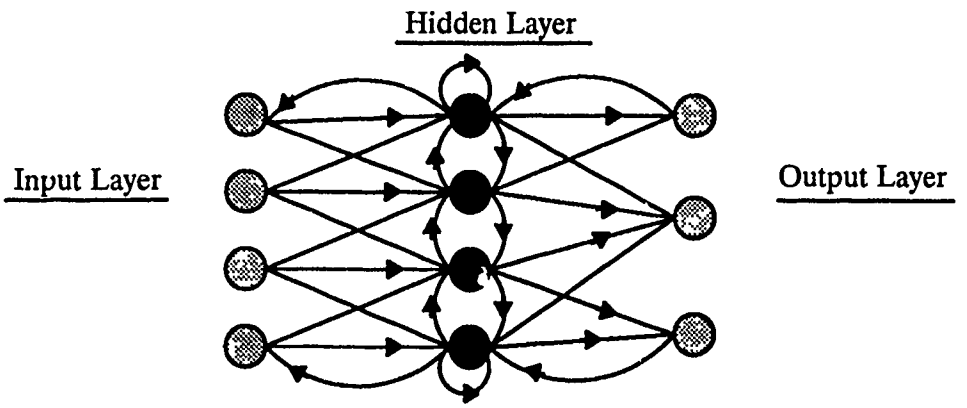


Figure 3.10 – Architecture of the Recurrent Neural Network

3.4.4 Haesloop and Holt (1990)

This article describes a Direct Linear Feedthrough Structure (DLF), a variation on the backpropagation network by adding direct connections from input layer to output layer and the application of this architecture on the process identification problem. In a three layer backpropagation network with one hidden layer, the connections from input layer to output layer represent a linear system,

whereas the remaining network models the nonlinear dynamics in the system. The general architecture is shown in Fig. 3.11.

The computational properties of the network are tested on the process identification of a surge tank that has a stream flowing into it at an independently determined rate and flowing out of the tank at a rate proportional to the square root of the height of fluid in the tank. When trained on the whole set of data, the DLF results in better learning accuracy and shorter learning time than the standard backpropagation network. The DLF architecture is also compared with a standard backpropagation network on the generalization capability by training on a training subset and testing on the remaining set of data. The DLF architecture shows better accuracy and extrapolation capability on the testing cases.

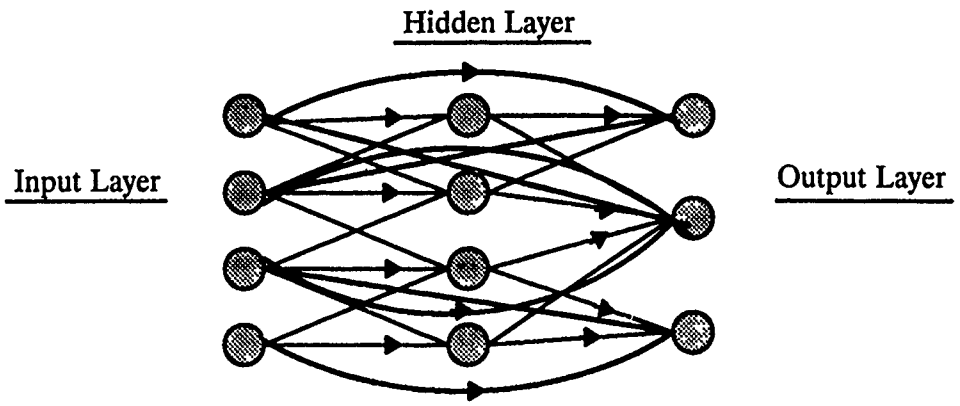


Figure 3.11 – Architecture of the Neural Network with Direct Linear Feedthrough

3.4.5 Hakim, et al. (1990)

Feedback mechanism is a necessity in building a neural network system with dynamic behavior. In nonlinear signal processing and time series prediction, dynamic recurrent network models would be better suited for studying chaotic and nonstationary time series. A new neural network architecture is proposed in this article to solve the system identification problem by introducing recurrent mechanism into the network through clustering and interconnections between nodes among different clusters in the hidden layer. Two neuron models, the classical neuron with graded response used by Hopfield and Tank, as well as a discrete-time model, are studied in detail. The parameters used for solving a system identification problem consist of the weight matrix of the middle neuron layer, the input and output connection matrices, the size and topology of the interconnection neighborhood, the neurons' time constants, and the shape of the nonlinearities. The architecture of the network is shown in Fig. 3.12.

The performance of the network is remarkable on two benchmark problems: the prediction of the logistic function with chaotic behavior and building a neural network Frequency Shift Keying (FSK) demodulator. Behaviors of the time parameters and the number of neurons in the middle layer are

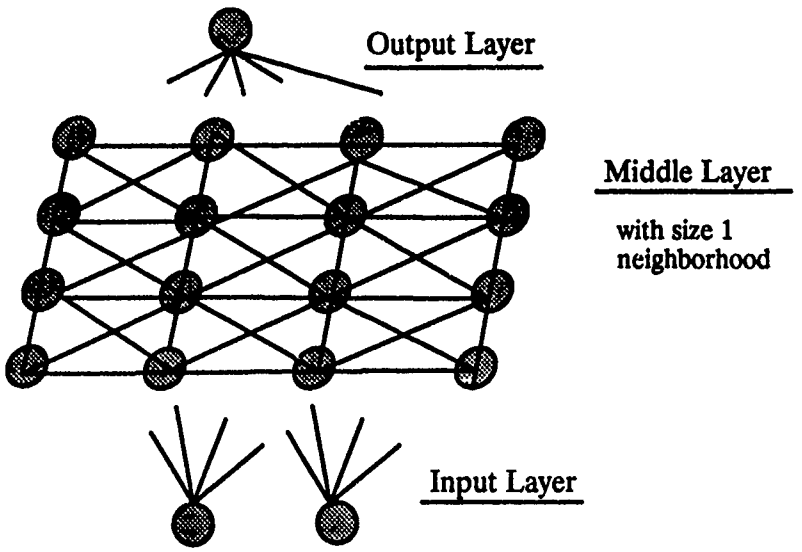


Figure 3.12 – Architecture of the Neural Network for System Identification

also investigated. One advantage of this architecture is that it can handle information from multi-channel measurements of the system.

3.4.6 Nishimura and Arai (1990)

This article describes a structured neural network and its application to power system state evaluation for proper power system control and operation (Fig. 3.13). The voltage measurement at different nodes of the power system is used to identify patterns of system performance. The new architecture consists of an input layer, a receptive layer, an associative layer, logic units and feedback mechanism as shown in Fig. 3.14. The network is constructed by wiring instead of learning, and the modeling and interpolation capability of the network is provided by the selective activation coefficients and the relative contribution coefficients.

The composition of each layer in the structured network is as follows. The input layer serves as the usual input terminals; the receptive layer is composed of several receptive strips that correspond to subpatterns in the input data; the associative layer integrates the outputs of the receptive layer; and the logic units process the information on the associative layer and generate the final output. The feedback mechanism changes the selective activation coefficients and the relative contribution coefficients according to some outputs of logic units, which also provides a deeper reasoning capability than a backpropagation network. The proposed network works very well on power system models.

3.4.7 Narendra and Parthasarathy (1990)

This article describes the application of neural networks in the identification and control of nonlinear dynamical systems. Multilayer backpropagation neural networks and recurrent networks are

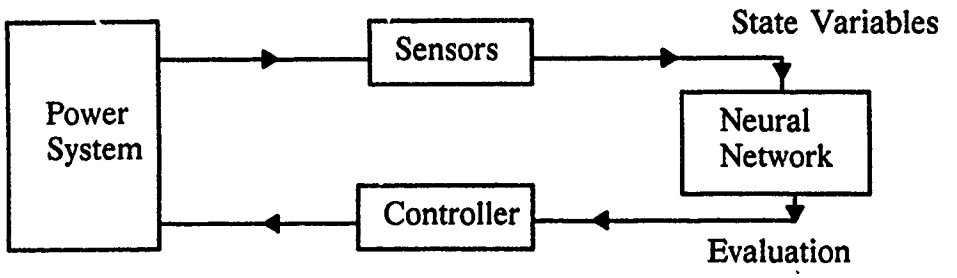


Figure 3.13 – Problem of Power System State Evaluation

treated in a unified fashion. Schemes for static and dynamic adjustment of parameters of neural networks are also proposed. Processes for direct adaptive control and indirect adaptive control of nonlinear systems using neural networks are presented with satisfactory simulation results.

Our coverage on the application of neural networks to system identification only provides a glimpse of this field due to time and space limitation. It is also of interest to notice that system identification via neural networks has not been seriously addressed for structural systems in civil engineering except by the short remarks made by Rehak, et al. (1989). Research will be excellent in this area and important in providing robust tools and insight for structural system analysis.

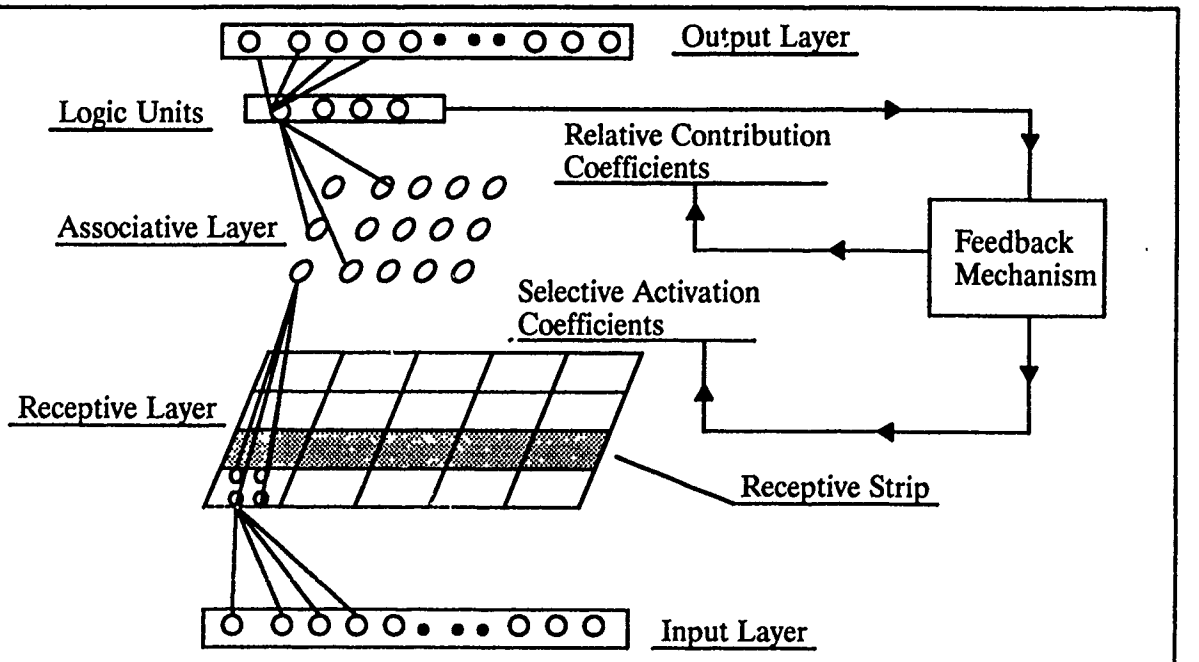


Figure 3.14 – Architecture of the Structured Neural Network

References

1. Fernandez, B., Parlos, A. G., and Tsai, W. K., "Nonlinear Dynamic System Identification Using Artificial Neural Networks," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17–21, 1990, II–133.

2. Garbor, D., Wilby, W. P., and Woodcock, R., "A Universal Nonlinear Filter, Predictor and Simulator Which Optimizes Itself by a Learning Process," *IEE Proc.*, 108B, pp. 422–438, 1961.

3. Haesloop, D., and Holt, B. R., "Neural Networks for Process Identification," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17–21, 1990, III–429.

4. Hakim, N. Z., Kaufman, J. J., Cerf, G., Meadows, H. E., "A Discrete-Time Neural Network Model for Systems Identification," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17–21, 1990, III–593.

5. Lapedes, A. S., and Farber, R. M., "Nonlinear Signal Processing Using Neural Networks: Prediction and System Modeling," *Technical Report LA–UR–87–2662*, Los Alamos National Laboratory, 1987.

6. Levin, E., Gewirtzman, R., and Inbar, G. F., "Neural Network Architecture for Adaptive System Modeling and Control," *Proceedings of the International Joint Conference on Neural Networks*, Washington, D. C., June 1989, II–311.

7. Narendra, K. S., and Parthasarathy, K., "Identification and Control of Dynamical Systems Using Neural Networks," *IEEE Transactions on Neural Networks*, Vol. 1, No. 1, pp. 4–27, March 1990.

8. Nishimura, K., and Arai, M., "Power System State Evaluation by Structured Neural Network," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17–21, 1990, I–271.

9. Rehak, D. R., Thewalt, C. R., and Doo, L. P., "Neural Network Approaches in Structural Mechanics Computation," *Proceedings of the Sessions Related to Computer Utilization at Structural Congress'89*, ASCE, pp. 168–176, May 1989.

10. Tishby, N., Levin, E., and Solla, S. A., "Consistent Inference of Probabilities in Layered Networks: Predictions and Generalization," *Proceedings of the International Joint Conference on Neural Networks*, Washington, D. C., June 1989, II–403.

11. Tenorio, M. F. M., and Lee, W.-T., "Self-Organizing Neural Network for the Identification Problem," *Advances in Neural Information Processing System I*, D. S. Touretzky, Ed., pp. 57–64, 1989.

12. Tenorio, M. F. M., and Lee, W.-T., "Self-Organizing Neural Network for Optimum Supervised Learning," *TR–EE–89–30*, School of Electrical Engineering, Purdue University, West Lafayette, Indiana, June 1989.

13. Ukrainec, A., Haykin, S., and McGregor, J., "A Neural Network Nonlinear Predictor," *Proceedings of the International Joint Conference on Neural Networks*, Washington, D. C., June 1989, II–622.

14. Zadeh, L. A., "From Circuit to System Theory," *Proc. IRE 50*, pp. 856–865, 1962.

3.5 Forecasting and Prediction

From the history of development in scientific research, it could not be over-emphasized that one of the central problems of science is to discover the underlying laws of the universe so that not only a natural phenomenon can be understood but also its future can be forecasted or predicted. It is also interesting to realize that the ability to give proper prediction bears a close tie with the quality of life, as are exemplified in weather forecasting and economic planning.

As pointed out by Weigent, et al. (1990), two types of knowledge are required in order to forecast the behavior of a natural system, namely, knowledge of the underlying laws and discovery of strong empirical regularities in observations of the system. Therefore, for the former, the prediction problem functions like an initial value problem which is fully determined by the differential equation and its initial conditions; for the latter, the behavior of the system is represented by its periodicity whether the periodicity is apparent or masked by noises. In other words, the system should be properly identified first. It is thus natural that system identification is the overture in producing the outcome of forecasting. Traditional prediction uses statistical methods such as curve fitting and regression analysis. Neural networks, due to their learning capability and intrinsic statistical characteristics, provide a potential tool and modeling methodology in this vital area of scientific endeavor.

Real world prediction problems range from bond rating (Dutta and Shekhar, 1988) and power system load forecasting (Atlas, et al., 1990) to sunspot activity, have been investigated by researchers from different disciplines. Due to their intrinsic properties, forecasting for problem domains with underlying principles or having well defined models is easier to model than that for nonconservative domains in which no well defined models exist. In the following paragraphs, some of the studies in this direction are described.

3.5.1 Farmer and Sidorowich (1987)

Though this article does not address the application of a neural network to prediction, it does influence the thinking and benchmark construction on neural network based prediction models. The article describes a forecasting technique specifically designed for chaotic data by embedding a time series in a state space using delay coordinates and modeling the nonlinear mapping using local approximation. The local approximation approach performs significantly better than the global approximation method introduced by Gabor, et al. (1960) and autoregressive models, on modeling the logistic map, the Mackey-Glass delay-differential equation, Taylor-Couette flow, and Rayleigh-Benard convection.

3.5.2 Dutta and Shekhar (1988)

This article describes the application of neural networks to the prediction of the ratings of corporate bonds, which belongs to the nonconservative problem domains where a domain model or theory is not well defined. For this problem, conventional mathematical modeling techniques such as statistical regression models have yielded poor results and it is difficult to build rule-based expert systems.

Feedforward networks with two and three layers were used in this study. Ten variables were selected as input units and the output unit represents the rating of the bond. Bond ratings and values of the financial variables for a set of industrial bonds issued by 47 companies were chosen at random as the data set and 30 of them were used for training and the remaining data for testing. The neural network-based models were compared with standard regression models on the accuracy of classification on each bond. To determine the minimal set of influential variables, six financial variables were also used for training and testing. The reported results indicate that neural network-based models outperformed the regression model by a considerable margin on both learned (92.3 vs. 61.5 percent) and testing cases (83.3 vs. 50 percent), and the three layer network gives best learning results. On the testing cases, the performance of two layer network is the same as the three layer network (83.3 percent). The reason is that the selected ten input financial features of a bond are relatively high level abstractions. On the other hand, the learning results with ten input variables give better performance than that with six input variables. One interesting result is that, on misclassified cases, the network prediction has at most one grade difference, whereas the regression model is often off by several ratings.

The success of the neural network-based models rests on the in-depth understanding of the problem and correct judgement on selecting the representing financial variables by the modelers. This in turn indicates the importance of the derivation of a good representation scheme.

3.5.3 Fozzard, Bradshaw, and Ceci (1989)

This article describes an application of a backpropagation network for daily solar flare forecasting and comparison of the network prediction with a rule-based expert system. An interesting feature of the approach is that it uses the identical representation scheme as used for the rule-based system. The architecture of the network is shown in Fig. 3.15. The 3 output units represent the 3 classes of solar flares to be forecasted, and the 17 input units provide a distributed coding of the 10 categories of input data that are used for the expert system. The network is trained and tested with data from the database of the expert system, and the performance of the network is at least as good as the expert system.

It should be pointed out that the network was only tested on a small segment of the 11-year solar cycle and no other representation scheme was investigated for the generalization analysis. However, this preliminary study does show the promise of neural networks in the field of forecasting where no underlying physical principle seems apparent at the moment.

3.5.4 Sharda and Patil (1990)

This article reports a comparative study of the forecasting capability of neural networks with conventional models based on Box-Jenkins Methods. A backpropagation network with similar architecture to NETtalk (Sejnowski and Rosenberg, 1986) is used to model the time series via time windows. The simulation results showed that both neural network models and conventional models performed equally well on the simulation problem so that neural networks can actually be used as a forecasting tool. The advantage of using neural networks is that it is a very simple model and is easy to build.

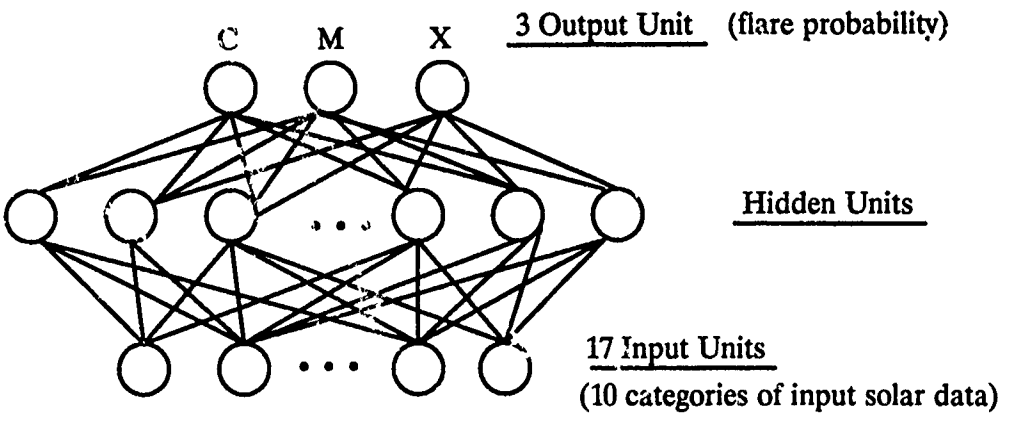


Figure 3.15 – Architecture of the Neural Network for Prediction

3.5.5 Walter, Ritter and Schulten (1990)

This article describes a new approach to the prediction of non-linear time sequence data, namely, the prediction of 3-D motion of an object in a set of nonlinear potentials of different orders of nonlinearity. The main idea is to use a Kohonen network to adaptively discretize the set of the input data, and to estimate in each discretization cell a separate set of linear prediction coefficients. A lattice with 12x12 neurons is used in this study. The network is trained with a series of trajectories with randomly chosen starting values and reasonable performance is obtained.

The unique feature of this approach is the use of a Kohonen network instead of the usual feedforward type network. It would be of interest in the field of prediction if an integrated system could be built with features from feedforward networks and self-organizing networks.

3.5.6 Weigend, Huberman and Rumelhart (1990)

This report describes the extension of feedforward networks utilized by Lapedes and Farber (1987) to predict future values of possibly noisy time series by extracting knowledge from the past. A three layer backpropagation network with one hidden layer (shown in Fig. 3.16) and its variation through a weight-elimination scheme and a time window representation scheme are used in modeling the behavior of the time varying system. The weight-elimination scheme is derived from a more complex cost function than the usual squared error cost function in that a cost measurement associated with each connection weight in the network is included. The issues of over-fitting and generalization capability while training a large network, and sigmoidal transfer function vs. radial basis functions, are analyzed numerically in the system training process. It was found that a densely trained network via weight-elimination scheme illustrated better generalization capability than sparse networks.

Two benchmark problems in statistics were tested on the forecasting capability of neural networks, namely, the sunspot activity prediction and the modeling of computational ecosystems. For the sun-

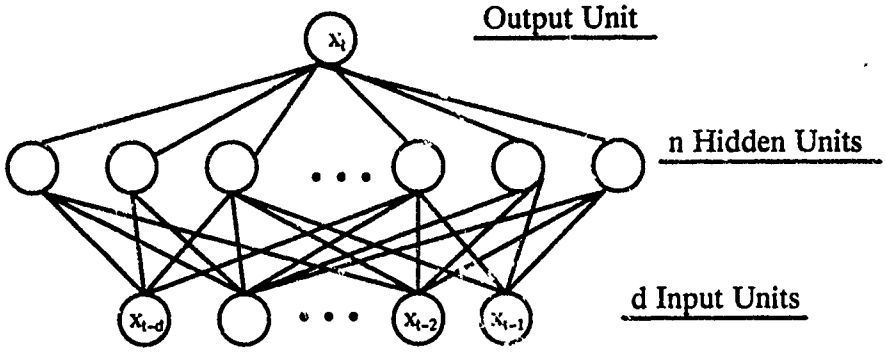


Figure 3.16 – Architecture of the Neural Network for Prediction

spot problem, the neural network model outperformed the threshold autoregressive model by Tong and Lim (1980), which is considered the best statistical model to date. For the prediction in a computational ecosystem, a time series for the use of resources is trained with a three layer backpropagation network with one hidden layer. Single and multi-step predictions of the ecosystem show good agreement between data and forecasting values. The eventual forecasting function after several million iterations exhibited a very similar frequency spectrum to the original data.

3.5.7 Werbos (1988)

It is perhaps fair to say that Werbos is one of the first few who applied neural networks to real world problems. This article describes a generalization of dynamic feedback in the backpropagation network to deal with recursive time-dependent networks and to use it in prediction, optimization over time and the analysis of the properties of a natural gas market model which has been used in a major study of natural gas deregulation.

3.5.8 White (1988)

This article describes the application of neural networks to an enticing field – stock market prediction. The objective of the work is to determine whether a neural network can decode previously undetected regularities in asset price movements, such as the daily fluctuations of common stock price, using the case of IBM daily common stock returns as an example. A sample of 1000 days of data were selected out of the available 5000 days of return data as a training set, together with samples of 500 days before and after the training period as testing cases. A three layer backpropagation network with one hidden layer is used in this study. Though results from this endeavor were not satisfactory, some valuable insights are worth mentioning: 1) modeling this highly nonconservative problem system is not easy with simple networks, 2) the simple network has the tendency to over-fit the price time series, and 3) the simple network is capable of extremely rich dynamic behavior.

References

1. Atlas, L., Cole, R., Connor, J., El-Sharkawi, M., Marks II, R., Muthusamy, Y., and Barnard, E., "Performance Comparisons Between Backpropagation Networks and Classification Trees on Three Real World Applications," in *Advances in Neural Information Processing System II*, D. S. Touretzky, Morgan Kaufmann Publaishers, Inc., pp. 622–629, 1990.

2. Dutta, S., and Shekhar, S., "Bond Rating: A Non-conservative Application of Neural Networks," *Proceedings of 1988 IEEE International Conference on Neural Networks*, II–443.

3. Farmer, J. D., and Sidorowich, J., "Predicting Chaotic Time Series," *Technical Report LA–UR–87–1502*, Los Alamos National Laboratory, 1987.

4. Fozzard, R., Bradshaw, G., and Ceci, L., "A Connectionist Expert System that Actually Works," in *Advances in Neural Information Processing System I*, D. S. Touretzky, Morgan Kaufmann Publishers, Inc., pp. 248–255, 1989.

5. Garbor, D., Wilby, W. P., and Woodcock, R., "A Universal Nonlinear Filter, Predictor and Simulator which Optimizes Itself by a Learning Process," *IEE Proc., 108B*, pp. 422–438, 1961.

6. Jones, R. D., Lee, Y. C., Barnes, C. W., Flake, G.W., Lee, K., Lewis, P. S., and Qian, S., "Function Approximation and Time Series Prediction with Neural Networks," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17–21, 1990, I–649.

7. Kamijo, K., and Tanigawa, T., "Stock Price Pattern Recognition: A Recurrent Neural Network Approach," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17–21, 1990, I–215.

8. Kimoto, T., Asakawa, K., Yoda, M., and Takeoka, M., "Stock Market Prediction System with Modular Networks," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17–21, 1990, I–1.

9. Lapedes, A. S., and Farber, R. M., "Nonlinear Signal Processing Using Neural Networks: Prediction and System Modeling," *Technical Report LA–UR–87–2662*, Los Alamos National Laboratory, 1987.

10. Levy, W. B., "Maximum Entropy Prediction in Neural Networks," *Proceedings of the International Joint Conference on Neural Networks*, Washington, D. C., 1990, I–7.

11. Odom, M. D., and Sharda, R., "A Neural Network Model for Bankruptcy Prediction," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17–21, 1990, II–163.

12. Rogers, D., "Statistical Prediction with Kanerva's Sparse Distributed Memory," in *Advances in Neural Information Processing System I*, D. S. Touretzky, Morgan Kaufmann Publishers, Inc., pp. 586, 1989.

13. Rogers, D., "Predicting Weather Using a Genetic Memory: A Combination of Kanerva's Sparse Distributed Memory with Holland's Genetic Algorithms," in *Advances in Neural Information Processing System II*, D. S. Touretzky, Morgan Kaufmann Publishers, Inc., pp. 455, 1990.

14. Saha, A., and Keeler, J. D., "Algorithms for Better Representation and Faster Learning in Radial basis Function Networks," in *Advances in Neural Information Processing System II*, D. S. Touretzky, Morgan Kaufmann Publishers, Inc., pp. 482, 1990.

15. Sejnowski, T. J., and Rosenberg, C. R., "NETtalk: A Parallel Network That Learns to Read Aloud," *The Johns Hopkins University Electrical Engineering and Computer Science Technical Report*, JHU/EECS-86/01, 1986.

16. Sharda, R., and Patil, R. B., "Neural Networks as Forecasting Experts: An Empirical Test," *Proceedings of the International Joint Conference on Neural Networks*, Washington, D. C., January 1990, II-491.

17. Surkan, A. J., and Singleton, J. C., "Neural Networks for Bond Rating Improved by Multiple Hidden Layers," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17-21, 1990, II-157.

18. Tishby, N., Levin, E., and Solla, S. A., "Consistent Inference of Probabilities in Layered Networks: Predictions and Generalization," *Proceedings of the International Joint Conference on Neural Networks*, Washington, D. C., June 1989, II-403.

19. Tong, H. and Lim, K. S., "Threshold Autoregression, Limit Cycles and Cyclical Data," *J. Royal Statistical Society*, B42, pp. 245, 1980.

20. Ukrainec, A., Haykin, S., and McGregor, J., "A Neural Network Nonlinear Predictor," *Proceedings of the International Joint Conference on Neural Networks*, Washington, D. C., June 1989, II- 622.

21. Villarreal, J., and Baffes, P., "Sunspot Prediction Using Neural Networks," *preprint*, Artificial Intelligence Section/FM72, Mission Planning and Analysis Division, NASA, Lyndon B. Johnson Space Center, Houston, TX, 1989.

22. Walter, J., Ritter, H., and Schulten, K., "Non-Linear Prediction with Self-Organizing Maps," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17-21, 1990, I-589.

23. Weigend, A. S., Huberman, B. A., and Rumelhart, D. E., "Predicting the Future: A Connectionist Approach," *Stanford-PDP-90-01*, Stanford University, April 1990.

24. Werbos, P., "Generalization of Backpropagation with Application to Recurrent Gas Market Model," *Neural Networks*, Vol. 1, pp. 339-356, 1988.

25. White, H., "Economic Prediction Using Neural Networks: The Case of IBM Daily Stock Return," *Proceedings of 1988 IEEE International Conference on Neural Networks*, II-451.

3.6 Control

In the control of a dynamic system, there usually exist two processes, namely, the learning or identification of the system and the extraction and enforcement of control signals. Control theory is a very well defined domain with much literature, and many learning methods used in neural network learning are closely related to methods that have been intensively studied in adaptive control theory. On the

other hand, because of the way a neural network operates and performs, the neural network based control has special characteristics that lack in traditional control theory. It has been illustrated in previous sections that neural networks are powerful tools in capturing the characteristics of a system through self-organization or learning. Because of this adaptive parameter estimation capability, neural networks are very well suited for application in engineering control.

To date, many kinds of neural networks have been studied for control applications. Like system identification, systems consisting of supervised learning models have found the most extensive usage in control. Backpropagation learning as well as reinforcement learning are considered to be the most suitable learning algorithm. One of the advantages of using reinforcement learning is that the learning process can be accomplished on-line. On the other hand, reinforcement learning provides flexibility in extracting control signals because it addresses the problem of improving performance as evaluated by any measure whose values can be supplied to the learning system (Barto, 1989).

The application of neural network-based control techniques has covered a broad range. Perhaps the most noteworthy one is the trucker backer-upper and broom balancer problems solved by Windrow and Nguyen (1987 and 1989). The majority of applications of neurocontrol are in all aspects of robot control and manufacturing process control. The following list of references gives a sketchy picture of applications in this field.

References

1. Aissi, C., and Chouikha, M. F., "A New Adaptive Control Algorithm Based on Signal Characterization," *Proceedings of the International Joint Conference on Neural networks*, San Diego, June 17-21, 1990, III-435.

2. Anderson, C. W., "Learning to Control An Inverted Pendulum Using Neural Networks," *IEEE Control Systems Magazine*, pp. 31-36, April 1989.

3. Baloch, A. A., "A Neural System for Behavioral Conditioning of Mobile Robots," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17-21, 1990, II-723.

4. Barto, A. G., Sutton, R. S., and Anderson, C. W., "Neuronlike Adaptive Elements That Can Solve Difficult Learning Control Problems," *IEEE Transactions On Systems, Man And Cybernetics*, Vol smc-13, No. 5, pp. 834-846, 1983.

5. Barto, A. G., "Connectionist Learning for Control: An overview," *COINS Technical Report 89-89*, Department of Computer and Information Science, University of Massachusetts, Sept. 1989.

6. Bassi, D. F., and Bekey, G. A., "High Precision Position Control By Cartesian Trajectory Feedback and Connectionist Inverse Dynamics Feedforward," *Proceedings of the International Joint Conference on Neural Networks*, Washington, D. C., 1989, II-325.

7. Cruse, H., and Bruwer, M., "A Simple Network Controlling the Movement of A Three Joint Planar Manipulator," in *Parallel Processing in Neural Systems and Computers*, R. Echmiller, G. Hartmann, G. Hauske, Editors, North-Holland, pp. 409, 1990.

8. Daunicht, W. J., "Neural Networks Mediating Linearizable Dynamic Redundant Sensory Motor Reflexes Characterized by Minimum of Hermitian Norm," *Proceedings of 1988 IEEE International Conference on Neural Networks*, II-611.

9. Daunight, W. J., and Werntges, H., "Optimal Solution of Under-determined Linear Matrix Equations in Neural Networks," in *Parallel Processing in Neural Systems and Computers*, R. Echmiller, G. Hartmann, G. Hauske, Editors, North-Holland, pp. 413, 1990.

10. Daunight, W. J., Lades, M., Werntges, H., and Eckmiller, R., "Inverse Kinematics with Obstacle Avoidance Implemented As A DEFAnet," in *Parallel Processing in Neural Systems and Computers*, R. Echmiller, G. Hartmann, G. Hauske, Editors, North-Holland, pp. 417, 1990.

11. Elsley, R. K., "A Learning Architecture for Control Based on Backpropagation Neural Network," *Proceedings of 1988 IEEE International Conference on Neural Networks*, II-587.

12. Fetz, E. E., Shupe, L. E., and Murthy, V. N., "Neural Networks Controlling Wrist Movement," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17-21, 1990, II-675.

13. Fukuda, T., Shibata, T., Tokita, M., and Mitsuoka, T., "Neural Network Application for Robotic Motion Control," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17-21, 1990, III-447.

14. Gardner, E. P., "What the Robot's Hand Should Tell the Robot's Brain: Feature Detection in A Biological Neural Network," *Proceedings of 1988 IEEE International Conference on Neural Networks*, I-557.

15. Graf, D. H., and LaLonde, W. R., "A Neural Controller for Collision-Free Movement of General Robot Manipulators," *Proceedings of 1988 IEEE International Conference on Neural Networks*, I-77.

16. Guez, A., Eilbert, J., and Kam, M., "Neuromorphic Architecture for Adaptive Robot Control: A Preliminary Analysis," *Proceedings of the First IEEE International Joint Conference on Neural Networks*, San Diego, 1987, IV-567.

17. Guez, A., "A Neuromorphic Controller With A Human Teacher," *Proceedings of the First International Conference on Neural Networks, IEEE*, San Diego, 1988, II-595.

18. Guez, A., and Ahmad, Z., "Solution to the Inverse Kinematics Problem in Robotics by Neural Networks," *Proceedings of 1988 IEEE International Conference on Neural Networks*, II-617.

19. Hakala, J., Steiner, R., Kreimeier, B., and Eckmiller, R., "Quasi-local Solution for Inverse Kinematics of A Redundant Robot Arm," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17-21, 1990, III-321.

20. Handelman, D. A., Lane, S. H., and Gelfand, J. J., "Integration of Knowledge Based System and Neural Network Techniques for Robot Control," *IFAC Workshop on Artificial Intelligence in Real-Time Control*, Swansea, 1988.

21. Hoptroff, R. G., Hall, T. J., and Burge, R. E., "Experiments with A Neural Controller," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17-21, 1990, II-735.

22. Hormel, M., "A Control Concept for a Multifingered Robot Gripper Based on Neuron-Like Associative Memory," in *Parallel Processing in Neural Systems and Computers*, R. Echmiller, G. Hartmann, G. Hauske, Editors, North-Holland, pp. 427, 1990.

23. Iberall, T., "A Ballpark Approach to Modelling Human Prehension," *Proceedings of the First IEEE International Joint Conference on Neural Networks*, San Diego, 1987, IV-535.

24. Iwata, T., Machida, K., and Toda, Y., "Fuzzy Control Using Neural Network Techniques," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17-21, 1990, III-365.

25. Jakubowicz, O., and Spina, R., "An Artificial Neural Network Approach for Solving Autonomous Navigation Control Problems," *Proceedings of The International Joint Conference on Neural Networks, Washington, D. C.*, 1990, II-367.

26. Josin, G., "Combinations of Neural Systems for Particular Application Situations," *Proceedings of the First IEEE International Joint Conference on Neural Networks*, San Diego, 1987, IV-517.

27. Josin, G., Chaney, D., and White, D., "Robot Control Using Neural Networks," *Proceedings of the First International Conference on Neural Networks, IEEE*, 1988, II, 625-632.

28. Josin, G., "Preliminary Development of A Neural network Autopilot Model for A High Performance Aircraft," *Proceedings of the International Joint Conference on Neural Networks*, Washington, D. C., 1990, II-547.

29. Khoukhi, A., "A Multilevel Neural Architecture For Robot Dynamic Control," *Proceedings of The International Joint Conference on Neural Networks, Washington, D. C.*, 1990, II-383.

30. Kim, Y. S., and Lee, J. G., "Design of A Pole-balancing Controller Using Neural Networks," *Proceedings of the International Joint Conference on Neural Networks*, Washington, D. C., 1989, II-619.

31. Kong, S. G., and Kosko, B., "Comparison of Fuzzy and Neural Truck Backer-upper Control Systems," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17-21, 1990, III-349.

32. Koza, J. R., and Keane, M. A., "Cart Centering And Broom Balancing By Genetically Breeding Populations Of Control Strategy Programs," *Proceedings of the First International Conference on Neural Networks, IEEE*, 1988, I-198.

33. Kumar, S. S., and Guez, A., "Adaptive Pole Placement For Neurocontrol," *Proceedings of The International Joint Conference on Neural Networks, Washington, D. C.*, 1990, II-397.

34. Kuperstein, M., and Rubinstein, J., "Implementation of An Adaptive Neural Controller For Sensory-motor Coordination," *Proceedings of the International Joint Conference on Neural Networks*, Washington, D. C., 1989, II-305.

35. LaLonde, W. R., and Maklin, M. B., "The Reactive Touch Model: Determining Which Way To Move To Avoid Touching," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17-21, 1990, II-741.

36. Lee, S., and Kil, R. M., "Robot Kinematic Control Based on Bidirectional Mapping Neural Network," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17–21, 1990, III–327.

37. Levin, E., Gewirtzman, R., Inbar, G. F., "Neural Network Architecture For Adaptive System Modeling And Control," *Proceedings of the International Joint Conference on Neural Networks*, Washington, D. C., 1989, II–311.

38. Liu, H., Iberall, T., and Bekey, G. A., "Building a Generic Architecture for Robot Hand Control," *Proceedings of 1988 IEEE International Conference on Neural Networks*, I–567.

39. Martinetz, T. M., Ritter, H. J., and Schulten, K. J., "3D–neural–net For Learning Visuomotor-coordination Of A Robot Arm," *Proceedings of the International Joint Conference on Neural Networks*, Washington, D. C., 1989, II–351.

40. Martinetz, T. M., and Schulten, K. J., "Hierarchical Neural Net for Learning Control of A Robot's Arm and Gripper," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17–21, 1990, II–747.

41. Nahvi, M. J., "On the Neural Network of A Synthetic Cerebellum to Control Fast Movements of An Arm," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17–21, 1990, II–765.

42. Nguyen, D., and Widrow, B., "The Trucker Backer–Upper: An Example of Self–Learning in Neural Networks," *Proceedings of the International Joint Conference on Neural Networks*, Washington, D. C., 1989, II–357.

43. Nguyen, L., Patel, R. V., and Khorasani, K., "Neural Network Architectures for the Forward Kinematics Problem in Robotics," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17–21, 1990, III–393.

44. Noos, F. A., and Lauria, F. E., "An Assembler for A Neural Network," *Proceedings of the First IEEE International Joint Conference on Neural Networks*, San Diego, 1987, IV–601.

45. Psaltis, D., Sideris, A., and Yamamura, A., "Neural Controllers," *Proceedings of the First IEEE International Joint Conference on Neural Networks*, San Diego, 1987, IV–551.

46. Patrikar, A., and Provence, J., "A Self-Organizing Controller for Dynamic Processes Using Neural Networks," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17–21, 1990, III–359.

47. Pellionisz, A. J., "Intelligent Decisions and Dynamic Coordination: Properties of Geometrical Representation by Generalized Frames Intrinsic to Neural and Robotic Systems," *Proceedings of 1988 IEEE International Conference on Neural Networks*, II–603.

48. Saxon, J. B., and Mukerjee, A., "Learning the Motion Map of A Robot Arm with Neural Networks," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17–21, 1990, II–777.

49. Scott, R. W., and Collins, D. J., "Neural Network Adaptive Controllers," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17–21, 1990, III–381.

50. Smith, T., Pitney, G. A., and Greenwood, D., "Application of Genetic Algorithm to A Neural Network For Finding Optimal Paths," *Proceedings of the First IEEE International Joint Conference on Neural Networks*, San Diego, 1987, IV-505.

51. Suddarth, S. C., Sutton, S. A., and Holden, D. C., "A Symbolic Neural Method for Solving Control Problems," *Proceedings of 1988 IEEE International Conference on Neural Networks*, I-515.

52. Swiniarski, R., "A Neuromorphic Cascaded Context Sensitive Controller for Robot Manipulator," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17-21, 1990, III-387.

53. Tolat, V. V., and Windrow, B., "An Adaptive 'Broom Balancer' With Visual Inputs," *Proceedings of the First International Conference on Neural Networks, IEEE*, 1988, II, 641-652.

54. Tolat, V. V., "An Adaptive 'Broom Balancer' With Visual Inputs," *Proceedings of the First International Conference on Neural Networks, IEEE*, San Diego, 1988, II-641.

55. Tsutsumi, K., and Matsumoto, H., "Neural Computation and Learning Strategy for Manipulator Position Control," *Proceedings of the First IEEE International Joint Conference on Neural Networks*, San Diego, 1987, IV-525.

56. Tsutsumi, K., Katayama, K., and Matsumoto, H., "Neural Computation For Controlling The Configuration Of 2-dimensional Truss Structure," *Proceedings of the First International Conference on Neural Networks, IEEE*, II, pp. 575-586, 1988.

57. Wang, G. J., and Miu, D. K., "Unsupervised Adaptive Neural-Network Control," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17-21, 1990, III-421.

58. Wang, S. D., and Yeh, H. M. S., "Self-Adaptive Neural Architectures for Control Applications," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17-21, 1990.

59. Werntges, H. W., "Delta Rule Based Neural Networks for Inverse Kinematics: Error Gradient Reconstruction Replaces the Teacher," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17-21, 1990, III-415.

60. Werntges, H. W., "Delta Rule Based Neural Networks May Be Applicable to a Control Task That Does Not Provide a Teacher," in *Parallel Processing in Neural Systems and Computers*, R. Echmiller, G. Hartmann, G. Hauske, Editors, North-Holland, pp. 435, 1990.

61. Widrow, B., "The Original Adaptive Neural Net Broom-balancer," *International Symposium On Circuits And Systems, IEEE*, pp. 351-357, May 1987.

62. Wilhelmsen, K., and Cotter, N. E., "Neural Network Based Controllers for a Single-Degree-of-Freedom Robotic Arm," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17-21, 1990, III-407.

63. Zak, S. H., "Robust Tracking Control of Dynamic Systems with Neural Networks," *Proceedings of the International Joint Conference on Neural Networks*, Washington, D. C., 1990, II-563.

3.7 Diagnostics Systems

Fault diagnosis of a system, due to its relation with pattern recognition, is probably one of the most suitable area for the application of neural networks. Many real world problems involve the use of detection such as the diagnosis of disease from symptoms or the diagnosis of engine malfunction through observation of engine behavior. The detection and diagnosis of faults can usually be processed in two steps: the recognition of abnormality in the system and the identification of causes for the abnormality or faults.

There are many approaches to the diagnosis of a system in general, such as the rule-based system, simulation-based expert system and neural network-based system. Though rule-based systems have been used successfully in many medical and engineering applications, the process of encoding knowledge in rules and the knowledge acquisition process are complicated and not easily achieved in a short time frame. For systems with the simulation of a physical system in the knowledge base, the simulation process is computationally intensive and time consuming. Because of the slow response time pertaining to an expert system-based approach to diagnosis, real-time application of those approaches would be too difficult to achieve. On the other hand, neural networks such as backpropagation networks, after proper training, will provide nearly real-time response for a diagnosis system in real world applications and the system can be developed in a short time.

The architecture of neural network diagnosis systems has been investigated by many researchers on different problems, and it can be of different forms which include the simple single backpropagation network, a hierarchy of different backpropagation networks such as that used in jet engine diagnosis (Dietz, et al., 1990), a hybrid system combining neural networks and rule-based systems (Tsoukalas and Reyes-Jimenez, 1990; Saito and Nakano, 1988; Schreinemakers and Touretzky, 1990), and a system incorporating supervised learning and self-organizing mechanisms, depending on the nature and complexity of the problem.

The application of neural networks to diagnose problems ranges from medical diagnosis and fault diagnosis in electrical systems to the maintenance and monitoring of chemical processes. In medically related diagnosis, several diseases have been studied such as the diagnosis of epilepsy through symptoms in a computer aided medical diagnosis system (Appolloni, el al., 1990), the diagnosis of disease of newborn babies through analysis of radiology images (Boone, et al., 1990), and the diagnosis of low back pain (Bounds, et al., 1988). A hybrid expert system incorporating a neural network for medical diagnosis has been proposed by Saito and Nakano (1988), and Schreinemakers and Touretzky (1990) proposed the use of OPS5 functions for the construction of a hybrid system for diagnosing mastitis in cows.

Perhaps the most noteworthy application is the neural network-based explosive detection system for safety checks at airports (Shea, Lin and Liu, 1989 and 1990). Dietz, et al. (1987, 1988 and 1989) constructs a real-time diagnosis system for failure detection in the bearings and the fuel system of a jet engine, and also a space shuttle engine system based on the test data. Casselman and Acres (1990) developed a comprehensive diagnosis system using several networks for the monitoring of a large satellite communication system. Neural networks have also found application in the fault diagnosis of electronic circuit boards (Kagle, et al., 1990), the automatic control system (Marko, et al., 1990), the operation of a nuclear plant model (Tsoukalas and Reyes-Jimenez, 1990), and in transportation engi-

neering by processing radar waves for detecting the presence of a waterproofing membrane in an asphalt covered bridge deck (Vrchovnik, et al., 1990).

In the following paragraphs, several typical applications are described in detail and relevant references are also provided.

3.7.1 Medical Applications

Appolloni, et al., (1990) describe an application in the diagnosis of epilepsy, a group of neurological disorders characterized by the recurrence of epileptic seizures. A data set has been constructed by collecting comprehensively all the clinical and laboratory information on 158 patients presenting epileptic symptoms. In one respect, the classification criterion on the disease has been proposed by the Commission on Classification and Terminology of the International League Against Epilepsy. The objective of the research is to compare the syndromic classification based on clinical criteria with the categorization achieved with a backpropagation network.

A three layer backpropagation network with one hidden layer is used in this study. The input layer has 724 units representing a list of questions coded in bit form, and the output has 31 nodes representing the 31 possible diagnoses. Besides, the 31 diagnoses can be clustered into 7 groups. There are totally 156 sets of data in which 134 sets correspond to reliable diagnoses and 22 sets to uncertain or fuzzy diagnoses. The former sets are used for training and the latter for testing the generalization capability of the trained network. Through trial and error, it was found that a hidden layer with 50 nodes gave the optimal results both in training and testing.

After training the network, the previous network is trimmed of connections and nodes with small connection strengths, and finally the network consists of 74 input units. This process distills the data representation scheme to an efficient form so that it gives about 80 percent of valid results on the single diagnosis and 95 percent on the clusters of diagnosis. Of course, a more powerful approach would be the use of a neural net-based expert system.

Boone, et al. (1990) used backpropagation neural networks for the interpretation of radiological images in computer aided medical diagnosis of certain diseases. There are two processes involved in the diagnosis of diseases based on radiological images: the abnormality identification of the images and the interpretation of the abnormal findings. For the identification of abnormal anatomical structures appearing on the radiographs, one hundred 25-pixel images, generated with Gaussian noise and with signals added to 50 percent of them, were used in the training of a three layer network to indicate abnormality in the image. The network thus has 25 nodes in the input layer, 5 nodes in the hidden layer, and 1 node in the output layer. For computer aided diagnosis, another backpropagation neural network is used to map the relationship between radiographic findings to a list of plausible diagnoses. It was decided that there were about 50 possible choices for radiographic findings, and 23 possible diagnoses to the newborn chest radiographs. The training and testing results showed close consistency with the diagnoses from doctors (79 percent of positive diagnosis and 99 of negative diagnosis).

3.7.2 Communication Systems

Casselman and Acres described a large system called DASA/LARS with extensive use of neural networks for the diagnosis on the operation and maintenance of satellite communication networks.

On-line sensor information in an operational environment is used for training, and the resulting neural network-based system has already been integrated into the working environment. The diagnosis is based on two kinds of inputs: data obtained from a swept-frequency spectrum analyzer and database information obtained from another subsystem in the Defense Satellite Communications Systems (DSCS). A fault is diagnosed through comparison of the observed spectral data with the planned parameters for each carrier stored in the DOSS database.

The backpropagation network is used in the system and 9 different architectures are constructed to diagnose a total of 13 different problems including the transponder saturation, data format problems such as wrong modulation, incorrect coding and transmitter filter malfunction, and the failure of an earth station's autotrack feature, etc. It is interesting that four layer backpropagation networks with two hidden layers are utilized for all the subsystems. The system has been tested on-line and has worked remarkably well. An updated system has been installed at the satellite operations center of DSCS at Fort Detrick, MD since 1989.

3.7.3 Mechanical Systems

Dietz, et al. (1987, 1988, and 1989) describe the fault diagnosis of jet and rocket engines by using neural networks to construct the mapping from patterns of sensor data to a pattern associated with a particular fault condition. Three layer backpropagation networks with one hidden layer are used in this study. A jet engine diagnostic system is built to identify the difference between behavior exhibited by bearing failures and that by fuel interruptions directly from sensor data. The architecture of the system is hierarchical and consists of five networks for each kind of sensor data. A higher level network is directly used to process the sensor data and recognize the fault type, and the two lower level networks are then trained to identify the severity and duration of the fault. The architecture for the process of data from a sensor is shown in Fig. 3.17.

For the current prototypic system, four sensors are employed to measure the combustion temperature, exhaust gas temperature, low pressure turbine rotational speed, and high pressure turbine rotational speed. Hence the system is totally composed of 20 networks and the design of the input layer is based on the sensor data acquired in a 4.0 second time interval. The training data are generated from an engine simulation program called ATEST. The testing of the trained network is performed by using crisp data and data generated with a certain percentage of noises. The training and testing of the system resulted in satisfying performance in jet engine diagnosis.

The article also describes another diagnosis system for the rocket engine used in the space shuttle, using experimental test data because of the lack of theoretical models. This later approach would have a more general implication for real-time diagnostics applications of neural networks.

3.7.4 Explosive Detection

Shea, et al. (1989 and 1990) describe the construction of a neural network-based system for explosive detection at airport check-ins. The application is probably one of the most successful cases and has generated lots of public interest in research and development of neural network technology. The

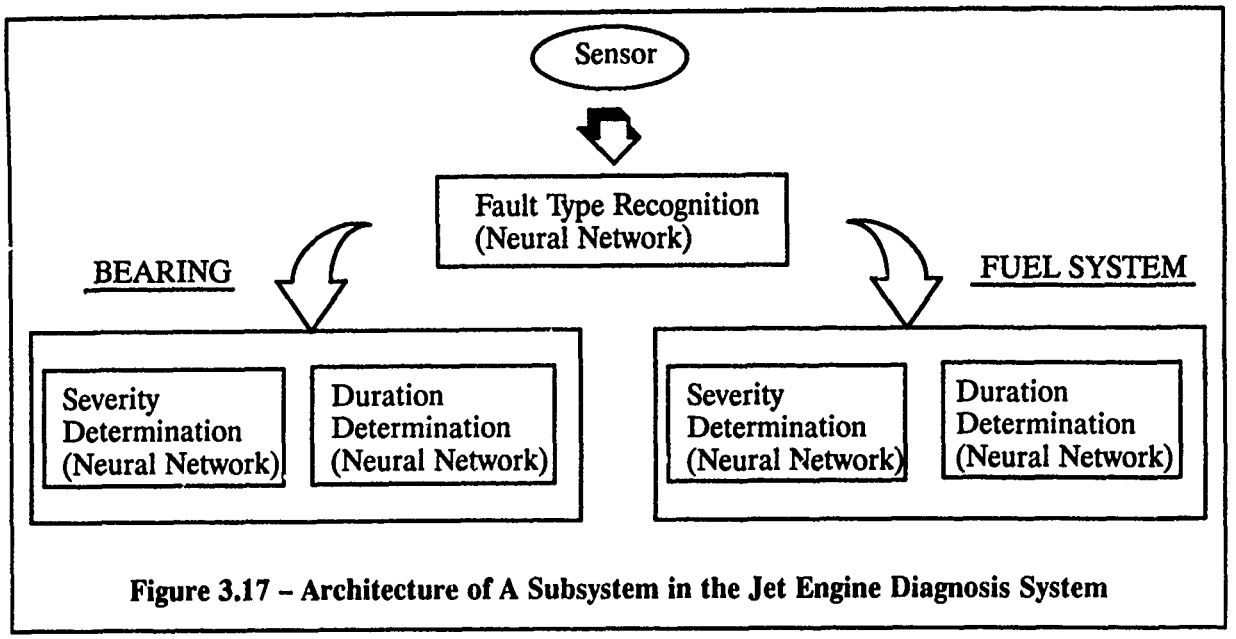


Figure 3.17 – Architecture of A Subsystem in the Jet Engine Diagnosis System

detection of explosives is based on the presence of nitrogen in the luggage using thermal neutron activation. A three layer backpropagation network with one hidden layer is used for training and testing the network from measurements gathered at the airport. The performance of the system is compared with a conventional system using standard statistical technique. The key parameters that measure the performance of the systems are the probability of detection (PD) for the minimum amount of explosive in a threat and the probability of false alarm (PFA) on bags without explosives. The neural network based system has been installed in several airports in parallel with the conventional system for a certain time. Through real world testing, it has been found that both systems perform equally well in terms of detection rate, and for false alarm rate the neural network based system is considerably better. Different learning models such as the counterpropagation network as well as the four layer backpropagation networks have also been investigated, and performance indicates that the three layer backpropagation network gives the best results.

References

1. Appolloni, B., Avanzini, G., CesanBianchi, N., and Ronchini, G., "Diagnosis of Epilepsy via Backpropagation," *Proceedings of the International Joint Conference on Neural Networks*, Washington, D. C., January 1990, II-571.

2. Becker, L. A., and Peng, J., "Network Processing of Hierarchical Knowledge for Classification and Diagnosis," *Proceedings of the First IEEE International Conference on Neural Networks*, San Diego, 1987, II-309.

3. Boone, J. M., Gross, G. W., and Shaber, G. S., "Computer Aided Radiological Diagnosis Using Neural Networks," *Proceedings of the International Joint Conference on Neural Networks*, Washington, D. C., January 1990, II-98.

4. Bounds, D. G., Lloyd, P.J., Mathew, B., and Waddell, G.,"A Multilayer Perceptron Network for the Diagnosis of Low Back Pain," *Proceedings of 1988 IEEE International Conference on Neural Networks*, II–481.

5. Casselman, F., and Acres, D. J., "DASA/LARS: A Large Diagnostic System Using Neural Networks," *Proceedings of the International Joint Conference on Neural Networks*, Washington, D. C., January 1990, II–539.

6. Dietz, W. E., and Ali, M., "Qualitative and Temporal Reasoning in Engine Behavior Analysis and Fault Diagnosis," *Proceedings of the Conference on Artificial Intelligence for Space Applications III*, Huntsville, AL, 1987.

7. Dietz, W. E., Kiech, E. L., and Ali, M., "Pattern Based Fault Diagnosis Using Neural Networks," *Proceedings of the International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems I*, Tullahoma, TN, 1988.

8. Dietz, W. E., Kiech, E. L., and Ali, M., "Neural Network Models Applied to Real Time Fault Diagnosis," *J. of Neural Network Computing*, Vol. 1, 1989.

9. Glover, D. E., "An Optical Fourier/Electronic Neurocomputer Automated Inspection System," *Proceedings of 1988 IEEE International Conference on Neural Networks*, I–569.

10. Hoskins, J. C., Kaliyur, K. M., Himmelblau, D. M., "Insipient Fault Detection and Diagnosis Using Artificial Neural Networks," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17–21, 1990, I–81.

11. Ishida, Y., "Fully Distributed Diagnosis by PDP Learning Algorithm: Towards Immune Network PDP Model," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17–21, 1990, I–777.

12. Kagle, B. J., Murphy, J. H., Koos, L. J., and Reeder, J. R., "Multi-fault Diagnosis of Electronic Circuit Boards Using Neural Networks," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17–21, 1990, II–197.

13. Kudrychi, T. P., *Neural Network Implementation of a Medical Diagnosis Expert System*, M. S. Thesis, Dept. of Electrical and Computer Engineering, University of Cincinnati, 1988.

14. Malkoff, D. B., "A Neural Network for Real-Time Signal Processing," in Advances in Neural Information Processing Systems 2, D. S. Touretzky, Ed., Morgan Kaufmann Publishers, pp. 248–257, 1990.

15. Marko, K. A., James, J., Dosdall, J., and Nurphy, J., "Automotive Control System Diagnostics Using Neural Nets for Rapid Pattern Classification of Large Data Sets," *Proceedings of the International Joint Conference on Neural Networks*, Washington, D. C., Vol. II, pp. 13–16, June 18–22, 1989.

16. Saito, Kazumi, and Nakano, Ryohei, "Medical Diagnostic Expert System Based On PDP Model," *Proceedings of the Second IEEE International Conference on Neural Networks*, San Diego, Vol. I, pp. 252–262, 1988.

17. Schreinemakers, J. F., and Touretzky, D. S., "Interfacing a Neural Network with a Rule-based Reasoner for Diagnosing Mastitis," *Proceedings of the International Joint Conference on Neural Networks*, Washington, D. C., January 1990, II–487.

18. Shea, P. M., and Lin, V., "Detection of Explosives in Checked Airline Baggage Using an Artificial Neural System," *Proceedings of the International Joint Conference on Neural Networks*, Washington, D. C., June 1989, II-31-34.

19. Shea, P. M., and Liu, F., "Operational Experience with a Neural Network in the Detection of Explosives in Checked Airline Luggage," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17-21, 1990, II-175.

20. Tan, A. H., Pan, Q., Lui, H. C., and Teh, H. H., "INSIDE: A Neuronet Based Hardware Fault Diagnostic System," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17-21, 1990, I-63.

21. Tenorio, M. F., "Extended Fuzzy Cognitive Maps Engines for the Diagnosis of Electro-mechanical and Electronic Equipment," *Proceedings of ISMM International Conference on Computer Applications in Design, Simulation and Analysis*, 1987.

22. Tsoukalas, L., and Reyes-Jimenez, J., "Hybrid Expert System: Neural-network Methodology for Nuclear Plant Monitoring and Diagnostics," *SPIE Vol. 1293, Applications of Artificial Intelligence VIII*, M. M. Trivedi, Editor, pp. 1024-1031, 1990.

23. Yamamoto, Y., and Venkatasubramanian, V., "Integrated Approach Using Neural Networks for Fault Detection and Diagnosis," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17-21, 1990, I-317.

24. Vrckovnik, G., Chung, T., and Carter, C. R., "An Application of Neural Networks to Impulse Radar Waveforms from Asphalt-Covered Bridge Decks," *Proceedings of the International Joint Conference on Neural Networks*, Washington, D. C., January 1990, II-453.

3.8 Planning, Scheduling, and Optimization

Planning and scheduling problems are highly constrained optimization problems or combinatorial optimization problem that is known to be NP-hard. The well known Traveling Salesman Problem is a good example involving path planning. The current approach to such problems is difficult if not impossible to find with an optimal or nearly-optimal solution. Usually, an admissible solution is good enough for acceptance. Since its advent, the Hopfield network has been considered by many researchers to obtain an optimal or nearly-optimal solution for this kind of NP-complete problem, and extensive research has been carried out on the solution to the Traveling Salesman Problem. Many real world problems, such as job-shop scheduling in mechanical engineering, crew scheduling in food service industry, and material handling, are combinatorial optimization problems and can be transformed into the frame of a Traveling Salesman Problem. Because of this, use of the Hopfield style network for planning and scheduling becomes feasible.

Though standard, Hopfield and Tank's network can be used to solve certain small optimization problems with constraints, the general use of this approach is impeded by its tendency to converge to the local minimum and its poor scaling properties for large problems. To overcome this difficulty, various modifications on the Hopfield and Tank network have been proposed. Some of the well known schemes are the Integer Linear Programming Neural Networks (Foo and Takefuji, 1988), the elastic net (Durbin and Willshaw, 1987), Supplier-Consumer Net (Parunak, et al., 1987), the Primal-Dual net-

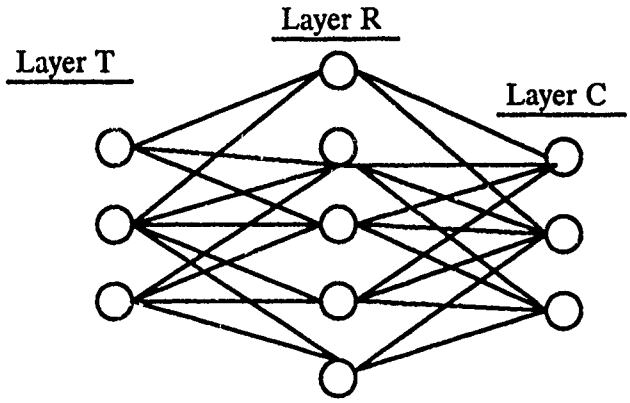


Figure 3.18 – Architecture of the Competitive Activation Based Network

work (Culioli, et al., 1990) and many others. After transforming a shared resource scheduling problem into an unshared resource problem, Bourret and Goodall (1989) proposed the competitive activation networks.

To date, different optimization problems encountered in various fields of disciplines have been solved with neural networks. Poliac, et al. (1987) used a novel representation scheme to solve the crew scheduling problem; Parunak, et al. (1987) proposed a supplier–consumer network to solve material handling problems; the Airline Marketing Tactician (Hutchison and Stephens, 1987) may be the first commercial application. For job–shop scheduling problem, Foo and Takefuji (1988), Chen (1990), and Zhou, et al. (1990), have investigated different approaches and obtained satisfactory results. Other applications include the time-table scheduling (Yu, 1990), object avoidance touring planning (Wong and Funka–Lea, 1990), linear programming (Culioli, et al., 1990; Kalaba and Moore, 1990), path optimization (Hassoun and Sanghvi, 1990), and the scheduling of satellite broadcasting times (Bourret, et al., 1990).

In the following paragraphs, some of the approaches to the scheduling and planning problems are described in detail to highlight the main features. There are also many publications on the theoretical analysis of optimization oriented neural networks (Maa and Shanblatt, 1990; Hellstrom and Kanal, 1990; Barbosa and de Carvalho, 1990). Due to time and space limitations, only some of the literature on that subject is included in the reference list.

3.8.1 Satellite Antennae Scheduling – Bourret and Goodall (1989)

The results reported by Bourret and Goodall (1989) are unique since it proposes and proves a theorem that transforms a shared resource scheduling problem into an unshared resource problem, and it introduces a competitive activation based neural network to solve the unshared resource scheduling problem. The proposed approach is tested on the optimal scheduling of antennae for low level satellites. The detailed antennae scheduling problem is, given the required broadcasting time, the

priority level of each satellite, and the time intervals within which the satellites are in sight of the various antennae, to optimize the total broadcasting time weighted by the priority of satellites.

The architecture of the competitive activation based neural network consists of three layers which are designated as layer R, layer C, and layer T, as is shown in Fig. 3.18. Each unit in layer R represents a time slice assigned to a satellite among those competing for that time slice, and the unit uses the competitive activation rules. Layer C consists of units that always keep the same activation level and represent each possible time slice. At the beginning of the computation, the strength of links between layer C and layer R is the given priority of the satellite. The last layer, Layer T, is composed of a number of units that correspond to the number of satellites in the system and the activation level of each unit represents how much broadcasting time has been scheduled for each satellite. A special competitive learning rule similar to the standard winner–takes–all scheme is proposed to determine the winner in layer R as well the modification of connection strengths or weights.

In another article, Bourret, et al. (1990) identifies the drawbacks encountered in the competitive activation based network for resource scheduling problem and presents a new implementation with a modified activation rule. The modified algorithm and architecture has the following features: 1) a new competitive output function is introduced to distribute the activation in layer C among competitors in layer R, 2) links are created between nodes in layer R to have competition with each other, 3) the decay factor is added to the activation rule for nodes in layer R, and 4) the activation update rule for nodes in layer R is modified to include a certain amount of noises. The modified scheme overcomes the shortcomings of the previous system and gives more robust results in different simulation problems on resource scheduling.

3.8.2 Robot Assembly Sequence Planning – Chen (1990)

Chen (1990) describes the theory and application of a Hopfield network to the solution of an assembly sequence problem which is also an AND/OR precedence-constrained traveling salesman problem. This problem involves the generation of all the possible assembly sequences and the determination of the most promising one. A modified Hopfield network is used to solve the planning problem. At first, the geometric constraints among parts are mapped into the connection weight matrix. The optimum assembly sequence can then be found through the ability of neurons' continuous dynamics adaptation to reach a lower energy measurement of the system.

In using Hopfield and Tank's approach to this combinatorial problem, binary threshold neurons and symmetric connection weight matrix are enforced. The precedence constraints can be mapped to the connection weight among neurons through giving the connection a positive or negative real value or using the property of biases to set the general level of excitability of the network such that the change in input–output relation at each neuron will result in a change of the activation level of the system. The optimum sequence with AND precedence relationship is solved using traditional second–order Hopfield network. A higher-order Hopfield network is also designed to solve the OR precedence relationship assembly sequences problem.

3.8.3 Linear Programming – Culioli, et al. (1990)

Culioli, et al. (1990) introduces a new Primal–Dual network to solve the general linear programming problem. The classical neural network approach to the linear programming problem is Hopfield and Tank's approach, and the solution usually converges to stable states. However, Hopfield and Tank's approach does not, in general, yield an optimal solution and it has poor scaling property. This shortcoming may come from the penalization treatment of the constraints. In the approach proposed, the constraints are treated with Lagrangian multipliers that converge to primal and dual admissible solution. It shows that the Primal–Dual network converges to admissible solutions, and can be used to get a very good approximation of the optimal cost.

3.8.4 Job-Shop Scheduling – Foo and Takefuji (1989)

Job-shop scheduling is a resource allocation problem involved with machines and the task jobs. Each job may also consist of several subjobs subject to precedence constraints. With this scheduling problem, it is very hard to obtain an optimal solution due to the large number of constraints. The Hopfield network and the Integer Linear Programming Neural Networks have been investigated for the job-shop scheduling problem, and it was observed that the two networks are not suitable for hardware implementation due to their poor scaling properties (Foo and Takefuji, 1988).

Foo and Takefuji (1988) are probably the first using the Hopfield type network to solve the job-shop scheduling problem. The approach proposed has a general use for all the NP–complete optimization problems with constraints. At first, the job–shop problem is mapped to a 2–D matrix representation of neurons similar to those for solving the traveling salesman problem. The constraints on operational precedence are imbedded in the network through application of constant positive and negative current biases to specific nodes. The solution of a job–shop problem is encoded in a set of cost function trees in the matrix of stable states. Each node in the set of trees represents a job, and each link represents the interdependency between jobs. The cost attached to each link is a function of the processing time of a particular job. The starting time of each job can be determined by traversing the parts leading to the root node of the tree. A computation circuit is used to compute the total completion times of all jobs, and the cost difference is added to the energy function of the Hopfield network. To reach the optimal solution, Simulated Annealing is used to help the system escape from local minimum. The use of an annealing algorithm is the most salient feature of the proposed approach.

The drawback associated with the use of an annealing algorithm is that it is computationally expensive because the procedure may take an infinite amount of time to find an optimal solution if the size of the problem grows larger and larger. This is also the criticism of using the Hopfield network for combinatorial optimization problems in general. The performance of the algorithm is tested by solving several job–shop scheduling problems with various degrees of complexity.

Foo and Takefuji (1988) also proposed an integer linear programming neural network based on a modified Tank and Hopfield neural network model by using linear measurement of the cost function to solve the job–shop scheduling problem. The cost function for minimization is the total starting times of all jobs subject to precedence constraints. The set of integer linear equations is solved by an iterative linear programming with integer adjustments technique, and the linear and nonlinear zero–one

variables are represented by linear sigmoid and nonlinear high-gain amplifiers with a response of a step function. The approach shows some improvement over the Hopfield network with simulated annealing.

Recently, Zhou, et al. (1990) introduced a novel approach to the job-shop scheduling problem by using a modified version of the Linear Programming Network described by Tank and Hopfield (1986). The proposed model uses a linear cost function instead of the quadratic energy function in Hopfield network. The important feature of the model is the incorporation of a product term into the energy constant function instead of using too many control variables to resolve conflicts of operation on the same machine. In a sense, it in fact implements the multiplication operation as an addition operation so that the resulting network has good scaling capability. On the other hand, in solving a simulation problem with 4/3 job-shop scheduling, the proposed network only uses a small fraction of neurons and connections of the regular Hopfield network or Integer Linear Programming Neural Networks (Foo and Takefuji, 1988).

3.8.5 Path Optimization - Hassoun and Sanghvi (1990)

Hassoun and Sanghvi propose a new neural network architecture for the path optimization problem in which a shortest path between two points in two or higher dimensions is sought. The architecture of the network is of multilayer modular form and the basic network consists of a locally-interconnected stage of simple neural subnets called comparators which perform node potential computations for a search map with one grounded node A. After certain computations are done at all nodes and the resulting collective computation leads to a stable potential surface having zero potential at the ground node A, an identical network is used to compute the second potential surface having another ground node B as the zero potential node. The nodes A and B are then assumed to be the end points of the optimal path. Next, the corresponding node potential pairs at A and B are added for nodes in the grid separately and each sum is compared to the minimum potential of the network using a final layer of input threshold neurons. The output of the final layer spatially encodes the optimal path between points A and B. The computation time of the network is determined by the speed at which the potential wave front spreads away from the ground node. In general, the convergence rate is very fast for the network. The performance of the algorithm is demonstrated through optimal path computation in 2-D space.

3.8.6 Time-Table Scheduling - Yu (1990)

Time-table scheduling is perhaps one of the classical problems in AI applications. Yu (1990) describes the application of a Hopfield network to the class scheduling in an educational institute. The scheduling problem is basically a graph coloring or graph partitioning problem. The relationship between the constraints and the time-slots can be represented by an edge-weighted graph which is very similar to the graph arisen from the decomposition problem converting from loosely synchronous problems to parallel machines. The Simulated Annealing algorithm is used with the optimization process to improve the capability of the network in escaping the local minimum. A problem of scheduling 64 classes into two time-slots is solved with the proposed approach.

References

1. Beckmann, J., "Resistive Network Approach for Obstacle Avoidance in Trajectory Planning," in *Parallel Processing in Neural Systems and Computers*, R. Echmiller, G. Hartmann, G. Hauske, Editors, North-Holland, pp. 405, 1990.

2. Barbosa, V. C., and de Carvalho, L. A. V., "Feasible Directions Linear Programming By Neural Networks," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17-21, 1990, III-941.

3. Bourret, P., and Goodall, S., "Optimal Scheduling by Competitive Activation: Application to the Satellite Antennae Scheduling Problem," *Proceedings of the International Joint Conference on Neural Networks*, Washington, D. C., 1989, I-565.

4. Bourret, P., Remy, F., and Goodall, S., "A Special Purpose Neural Network for Scheduling Satellite Broadcasting Times," *Proceedings of the International Joint Conference on Neural Networks*, Washington, D. C., 1990, II-535.

5. Chen, C. L. P., "Neural Computation for Planning AND/OR Precedence-Constraint Robot Assembly Sequences," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17-21, 1990, I-127.

6. Culioli, J.-C., Protopopescu, V., Britton Jr., C. L., and Ericson, M. N., "Neural Networks Models for Linear Programming, " *Proceedings of the International Joint Conference on Neural Networks*, Washington, D. C., 1990, I-293.

7. Denton, J. W., Madey, G. R.,"Impact of Neurocomputing on Operation Research," in *Impact of Recent Computer Advances on Operation Research*, Elsevier Science Publishing Co., Inc., New York, pp. 302-313, 1989.

8. Durbin, R., and Willshaw, D. J., "An Analogue Approach to the Traveling Salesman Problem Using an Elastic Net Method," *Nature* 326, pp. 689-691, 1987.

9. Fang, L., Wilson, W. H., and Li, T., "A Neural Network for Job Sequencing," in *Parallel Processing in Neural Systems and Computers*, R. Echmiller, G. Hartmann, G. Hauske, Editors, North-Holland, pp. 253, 1990.

10. Foo, Y. S., and Takefuji, Y., "Stochastic Neural Networks for Solving Job-shop Scheduling: Part I: Problem Presentation," *Proceedings of 1988 IEEE International Conference on Neural Networks*, II-275.

11. Foo, Y. S., and Takefuji, Y., "Stochastic Neural Networks for Solving Job-shop Scheduling: Part II: Architecture and Simulations," *Proceedings of 1988 IEEE International Conference on Neural Networks*, II-283.

12. Foo, Y. S., and Takefuji, Y., "Integer Linear Programming Neural Networks for Job-Shop Scheduling," *Proceedings of 1988 IEEE International Conference on Neural Networks*, II-341.

13. Greenberg, H. J., "Neural Networks for An Intelligent Mathematical Programming System," in *Impact of Recent Computer Advances on Operation Research*, Elsevier Science Publishing Co., Inc., New York, pp. 314-320, 1989.

14. Gulati, S., Lyengar, S. S., Toomarian, N., Protopopescu, V., and Barhen, J., "Nonlinear Neural Networks for Deterministic Scheduling," *Proceedings of the First International Conference on Neural Networks*, San Diego, IEEE, 1987, IV-745.

15. Hassoun, M. H., and Sanghvi, A., "Localiy Interconnected Layered Neural Network for Path Optimization," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17-21, 1990, III-811.

16. Hellstrom, B. J., and Kanal, L. N., "The Definition of Necessary Hidden Units in Neural Networks for Combinatorial Optimization," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17-21, 1990, III-775.

17. Helton, R., "A Self-Adaptive Connectionist Shortest-Path Algorithm Utilizing Relaxation Methods," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17-21, 1990, III-895.

18. Hemani, A., and Postula, A., "Scheduling by Self-Organization," *Proceedings of the International Joint Conference on Neural Networks*, Washington, D. C., 1989, II-543.

19. Ho, A. W., and Fox, G. C., "Parallel Neural Net Path Planner in Hypercubes and Transputers," in *Parallel Processing in Neural Systems and Computers*, R. Echmiller, G. Hartmann, G. Hauske, Editors, North-Holland, pp. 421, 1990.

20. Hutchison, W. R., and Stephens, K. R., "The Airline Marketing Tactisian (AMT): A Commercial Application of Adaptive Networking," *Proceedings of the First International Conference on Neural Networks*, San Diego, IEEE, 1987, IV-753.

21. Kalaba, R., and Moore, J. E., "Linear Programming and Associative Memory Matrices," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17-21, 1990, III-883.

22. Kamgar-Parsi, B., and Gualtieri, J. A., "Solving Inversion Problems with Neural Networks," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17-21, 1990, III-955.

23. Maa, C. Y., and Shanblatt, M. A., "Stability of Linear Programming Neural Network for Problems with Hypercube Feasible Region," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17-21, 1990, III-759.

24. Masti, C. L., and Livingston, D. L., "Neural Networks for Addressing the Decomposition Problem in Task Planning," *Proceedings of the International Joint Conference on Neural Networks*, Washington, D. C., 1989, II-555.

25. Matsuda, S., and Akimoto, Y., "The Representation of Large Numbers in Neural Networks and Its Application to Economical Load Dispatching of Electric Power," *Proceedings of the International Joint Conference on Neural Networks*, Washington, D. C., 1989, I-587.

26. Parunak, H. V. D., Kindrick, J., and Irish, B. W., "Supplier-Consumer Nets for Material Handling," *Proceedings of the First International Conference on Neural Networks*, San Diego, IEEE, 1987, IV-787.

27. Pitney, G., Smith, T. R., and Greenwood, D., "Genetic Design of Processing Elements for Path Planning Networks," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17-21, 1990, III-925.

28. Poliac, M. O., Lee, E. B., Slagle, J. R., and Wick, M. R., "A Crew Scheduling Problem," *Proceedings of the First International Conference on Neural Networks*, San Diego, IEEE, 1987, IV-779.

29. Reibling, L. A., and Olinger, M. D., "A Neural Network Implementation of Parallel Search for Multiple Paths," *Proceedings of the International Joint Conference on Neural Networks*, Washington, D. C., 1990, II-675.

30. Shepanski, J. F., and Macy, S. A., "Teaching Artificial Neural Systems to Drive: Manual Training Techniques for Autonomous Systems," in *Neural Information Processing Systems*, D. Z. Anderson, Editor, American Institute of Physics, New York, pp. 693-700, 1988.

31. Wang, J., and Malakooti, B., "Artificial Neural Networks for Multiple Criteria Decision Making," *Proceedings of the International Joint Conference on Neural Networks*, Washington, D. C., 1990, II-495.

32. Whitehead, S. D., and Dyer, M. G., "Connectionist Designs on Planning," *Proceedings of the 1988 Connectionist Models, Summer School*, D. Touretzky, G. Hinton and T. Sejnowski (Eds.), pp. 356-370, Carnegie Mellon University, Morgan Kaufman Publishers, Inc., 1989.

33. Wong, W. S., and Funka-Lea, C. A., "An Elastic Net Solution To Obstacle Avoidance tour Planning," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17-21, 1990, III-799.

34. Wu, C. H., Whitson, G. M., Montllor, G. J., "PROCANS: A Protein Classification System Using A Neural Network," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17-21, 1990, II-91.

35. Yu, T. L., "Time-table Scheduling Using Neural Network Algorithms," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17-21, 1990, I-279.

36. Yue, T. W., and Fu, L. C., "Ineffectiveness in Solving Combinatorial Optimization Problems Using A Hopfield Network: A New Perspective from Aliasing Effect," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17-21, 1990, III-787.

37. Zhou, D. N., Cherkassky, V., Baldwin, T. R., and Hong, D. W., "Scaling Neural Network for Job-Shop Scheduling," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17-21, 1990, III-889.

3.8.7 On the Theory of Optimization

Previous descriptions on solving planning and scheduling problems with neural networks have illustrated that these are optimization problems. There has been a lot of work on the theoretical analysis of using neural networks to combinatorial optimization problems especially on the Traveling Salesman Problem. The following list is provided for the purpose of completeness of presentation and a critical examination on the theoretical analysis will provide insight and new direction on solving real world optimization problems.

References

1. Abe, S., "Theories on the Hopfield Neural Networks," *Proceedings of the International Joint Conference on Neural Networks*, Washington, D. C., 1989, I-557.

2. Adorf, K. M., and Johnston, M. D., "A Discrete Stochastic Neural Network Algorithm for Constraint Satisfaction Problems," *Proceedings of the International Jointe on Neural Networks*, San Diego, June 17-21, 1990.

3. Akiyama, Y., Yamashita, A., Kajuura, M., and Aiso, H., "Combinatorial Optimization with Gaussian Machines," *Proceedings of the International Joint Conference on Neural Networks*, Washington, D. C., 1989, I-533.

4. Atkins, M., "Sorting by Hopfield Net," *Proceedings of the International Joint Conference on Neural Networks*, Washington, D. C., 1989, II-65.

5. Baum, E. B., "Towards Practical Neural Computation for Combinatorial Optimization Problems," in *Neural for Computing*, J. S. Denker, Editor, AIP Proceedings 151, American Institute of Physics, pp. 47-52, 1986.

6. Bibro, G., Mann, R., Miller, T. K., Snyder, W. E., Van den Bout, D. E., and White, M., "Optimization By Mean Field Annealing," *Advances in Neural Information Processing Systems I*, D. S. Touretzky, Editor, Morgan Kaufmann Publishers, Inc., pp. 91, 1989.

7. Bruck, J., and Goodman, J. W., "A Generalized Convergence Theorem for Neural Networks and Its Applications in Combinatorial Optimization," *Proceedings of the First International Conference on Neural Networks, IEEE*, 1987, III-649.

8. Bourret, P., and Goodall, S., "Optimal Scheduling by Competitive Activation: Application to the Satellite Antennae Scheduling Problem," *Proceedings of the International Joint Conference on Neural Networks*, Washington, D. C., 1989, I-565.

9. Cervantes, J. H., and Hildebrant, R. R., "Comparison of Three Neuron-based Computation Scheme," *Proceedings of the First IEEE International Conference on Neural Networks*, San Diego, 1987, III-657.

10. Chen, S., Xu, X., Tsai, W. T., and Huang, N.-K., *A Case Study of Solving Optimization Problems Using Neural Network*, Department of Computer Science and Electrical Engineering, University of Minnesota, 1988.

11. Cooper, L., "Optimization of Neural Network Connectivity," *Proceedings of the First IEEE International Conference on Neural Networks*, San Diego, 1987, III-767.

12. Dahl, E. D., "Neural Network Algorithm for an NP-complete Problem: Map and Graph Coloring," *Proceedings of the First IEEE International Joint Conference on Neural Networks*, San Diego, 1990, III-113.

13. Dodd, N., "Optimization of Network Structure Using Genetic Techniques," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17-21, 1990.

14. Dress, W. B., "Darwinian Optimization of Synthetic Neural Systems," *Proceedings of the First International Conference on Neural Networks, IEEE*, 1987, III-769.

15. Foo, Y. P. S., "Solving Large-scale Optimization Problems by Divide-and-conquer Neural Networks," *Proceedings of the International Joint Conference on Neural Networks*, Washington, D.C., 1989, I-507.

16. Gardner, S., "Statistical Dynamics of Ultradiffusion in Hierarchical Systems," *Proceedings of the First IEEE International Conference on Neural Networks*, San Diego, 1987, III-759.

17. Gaspin, C., "Automatic Translation of Constraints for Solving Optimization Problems By Neural Networks," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17-21, 1990.

18. Goldstein, M., Toomarian, N., and Barhen, J., "A Comparison Study of Optimization Methods for the Bipartite Matching Problem (BMP)," *Proceedings of 1988 IEEE International Conference in Neural Networks*, II-267.

19. Guha, A., and Mathur, A., "Setpoint Control Based on Reinforcement Learning," *Proceedings of the International Joint Conference on Neural Networks*, Washington, D. C., 1990, II-511.

20. Gutzmann, K. M., "Combinatorial Optimization Using a Continuous State Boltzmann Machine," *Proceedings of the First IEEE International Conference on Neural Networks*, San Diego, 1987, III-721.

21. Hassoun, M. H., and Sanghvi, A., "Locally Interconnected Layered Neural Network for Path Optimization," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17-21, 1990.

22. Hegde, S. U., Street, J. L., and Levy, W. B., "Determination of Parameters in a Hopfield-Tank Computational Network," *Proceedings of 1988 IEEE International Conference on Neural Networks*, II-291.

23. Hopfield, J. J., "Neural Networks and Physical Systems with Emergent Collective Computational Abilities," *Proceedings of the National Academy of Science*, Vol. 79, pp. 2554-2558, 1982.

24. Hopfield, J. J., and Tank, D. W., "Computing with Neural Circuits: A Model," *Science*, Vol. 233, pp. 625-633, 1985.

25. Hopfield, J. J., and Tank, D. W., "Neural Computation of Decision in Optimization Problems," *Biological Cybernetics*, Vol. 52, pp. 141-152, 1985.

26. Jeffrey, W., Rosner, R., "Neural Network Processing as a Tool for Function Optimization," in *Neural networks for Computing*, J. S. Denker, Editor, AIP Proceedings 151, American Institute of Physics, pp. 241-246, 1986.

27. Kamgar-Parsi, B., and Kamgar-Parsi, B., "An Efficient Model of Neural Networks for Optimization," *Proceedings of the First International Conference on Neural Networks, IEEE*, 1987, III-785.

28. Lee, B. W., and Sheu, B. J., "Combinatorial Optimization Using Competitive-Hopfield Neural Networks," *Proceedings of the International Joint Conference on Neural Networks*, Washington, D. C., 1990, II-627.

29. Leong, H. M., "Optimization with Artificial Neural Network Systems: A Mapping Principle and a Comparison to Gradient Based Methods," in *Neural Information Processing Systems*, D. Z. Anderson, Editor, American Institute of Physics, pp. 474-484, 1988.

30. Levy, B. C., and Adams, M. B., "Global Optimization with Stochastic Neural Networks,"*Proceedings of the First International Conference on Neural Networks, IEEE*, 1987, III-681.

31. Marcus, R., "A Connectionist Algorithm for Minimum Cost Network Flows," *Proceedings of the First IEEE International Conference on Neural Networks*, San Diego, 1987, III-735.

32. Matsuba, I., "Optimal Simulated Annealing Method and Its Application to Combinatorial Problems," *Proceedings of the International Joint Conference on Neural Networks*, Washington, D. C., 1989, I-541.

33. Matsuda, S., and Akimoto, Y., "The Representation of Large Numbers in Neural Netwc-lrc and Its Application to Economical Load Dispatching of Electric Power," *Proceedings of the International Joint Conference on Neural Networks*, Washington, D. C., 1989, I-587.

34. Matsuyama, Y., "Competitive Self-organization and Combinatorial Optimization: Applications to Traveling Salesman Problem," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17-21, 1990.

35. Mjolsness, E., Garrett, C., and Miranker, W. L., "Multiscale Optimization in Neural Nets: Preliminary Report," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17-21, 1990.

36. Moopenn, A., Thakoor, A. P., and Duong, T., "A Neural Network for Euclidean Distance Minimization," *Proceedings of 1988 IEEE International Conference in Neural Networks*, II-349.

37. Naft, J., "Neuropt: Neurocomputing for Multiobjective Design Optimization for Printed Circuit Board Component Placement," *Proceedings of the International Joint Conference on Neural Networks*, Washington, D. C., 1989, I-503.

38. Parks, M., "Characterization of the Boltzmann Machine Learning Rate," *Proceedings of the First IEEE International Conference on Neural Networks*, San Diego, 1987, III-715.

39. Platt, J. C., and Barr, A. H., "Constrained Differential Optimization,"in *Neural Information Processing Syste.ns*, D. Z. Anderson, Editor, American Institute of Physics, pp. 364-369, 1988.

40. Protzel, P. w., "Comparative Performance Measure for Neural Networks Solving Optimization Problems," *Proceedings of the International Joint Conference on Neural Networks*, Washington, D. C., 1990, II-523.

41. Ramanujam, J., and Sadayappan, P., "Optimization by Neural Networks," *Proceedings of 1988 IEEE International Conference in Neural Networks*, II-325.

42. Roysam, B., and Miller, M. I., "Mapping Deterministic Rules to Stochastic Representation Vis Gibbs' Distributions on Massively Parallel Analog Networks: Application to Global Optimization," in *Proceedings of 1988 Connectionist Models Summer School*, D. Touretzky, G. Hinton and T. Sejnowski, Eds., san Mateo, CA: Morgan Kaufmann, pp. 229-238, 1989.

43. Tagliarini, G. A., and Page, E. W., "Solving Constraint Satisfaction Problems with Neural Networks," *Proceedings of the First IEEE International Conference on Neural Networks*, San Diego, 1987, III-741.

44. Tagliarini, G. A., and Page, E. W., "Learning in Systematically Designed Networks," *Proceedings of the International Joint Conference on Neural Networks*, Washington, D. C., 1989, I-497.

45. Takefuji, Y., Lee, K. C., and Tanaka, T., "A Two Step Sorting Algorithm," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17-21, 1990.

46. Thomae, D. A., and Van En Bout, D. E., "Encoding Logical Constraints into Neural Network Cost Functions," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17-21, 1990.

47. Yau, H. C., and Manry, M. T., "Sigma-Pi Implementation of a Gaussian Classifier," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17-21, 1990.

48. Zak, M., Toomarian, N., and Barhen, J., "Creative Dynamics Approach to Optimization Problems," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17-21, 1990.

3.8.8 On the Traveling Salesman Problem

Some references directly related to the solution of the Traveling Salesman Problem are listed below to provide some pointers for those who are specifically interested in this problem.

References

1. Angeniol, B., De La Croix Vaubois, G., and Le Texier, J.-Y., "Self-Organizing Feature Maps and the Traveling Salesman Problem," *Neural Network*, Vol. 1, pp. 289-293, 1988.

2. Bagherzadeh, N., Kerola, T., Leddy, B., and Brice, R., "On Parallel Execution of the Traveling Salesman Problem on a Neural Network Model," *Proceedings of the First International Conference on Neural Networks, IEEE*, 1987, III-317.

3. Brandt, R. D., Wang, Y., Laub, A. J., and Mitra, S. K., "Alternative Networks for Solving the Traveling Salesman Problem and the List-Matching Problem," *Proceedings of 1988 IEEE International Conference on Neural Networks*, II-333.

4. Burr, D. J., "An Improved Elastic Net Method for Traveling Salesman Problem," *Proceedings of 1988 IEEE International Conference on Neural Networks*, I-69.

5. Durbin, R., Willshaw, D. J., "An Analogue Approach to the Traveling Salesman Problem Using an Elastic Net Method," *Nature*, Vol. 326, pp. 689-691.

6. Heuter, G. J., "Solution of the Traveling Salesman Problem with an Adaptive Ring," *Proceedings of 1988 IEEE International Conference on Neural Networks*, I-85.

7. Jeong, C. S., Kim, M. H., "Fast Parallel Simulated Annealing for Traveling Salesman Problem," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17-21, 1990.

8. Joppe, A., Cardon, H. R. A., and Bioch, J. C., "A Neural Network for Solving the Traveling Salesman Problem on the Basis of City Adjacency in the Tour," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17-21, 1990.

9. Kadaba, N., Nygard, K. E., Juell, P. L., and Kangas, L., "Modular Backpropagation Neural Networks for Large Domain Pattern Classification," *Proceedings of the International Joint Conference on Neural Networks*, Washington, D.C., January 1990, II-551.

10. Kahng, A. B., "Traveling Salesman Heuristics and Embedding Dimension in the Hopfield Model," *Proceedings of the International Joint Conference on Neural Networks*, Washington, D.C., 1989, I-513.

11. Lawler, E. L., Lenstra, J. K., Rinnooy Kan, A. H. G., and Shmoys, D. B., *The Traveling Salesman Problem*, Wiley, New York, 1986.

12. Lin, S., Kernighan, B. W., "An Effective Heuristic Algorithm for the Traveling Salesman Problem," *Operation Research*, Vol. 21, pp. 498-516.

13. Lister, R., "Segment Reversal and the Traveling Salesman Problem," *Proceedings of the International Joint Conference on Neural Networks*, Washington, D.C., January 1990, I-424.

14. Morita, A., Imai, Y., Noda, A., and Takegaki, M., "Fuzzy Knowledge Model of Neural Network type: A Model Which Can Be Refined By Learning," *Proceedings of the International Joint Conference on Neural Networks*, Washington, D.C., January 1990, II-55.

15. Protzel, P. W., and Arras, M. K., "Fault-Tolerance of Optimization Networks: Treating Faults as Additional Constraints," *Proceedings of the International Joint Conference on Neural Networks*, Washington, D.C., January 1990, I-455.

16. Szu, H., "Fast TSP Algorithm Based on Binary Neuron Output and Analog Neuron Input Using the Zero-Diagonal Interconnection Matrix and Necessary and Sufficient Constraints of the Permutation Matrix," *Proceedings of 1988 IEEE International Conference on Neural Networks*, II-259.

17. Szu, H., "Colored Noise Annealing Benchmark by Exhaustive Solutions of TSP," *Proceedings of the International Joint Conference on Neural Networks*, Washington, D.C., January 1990, I-317.

18. Szu, H., and Foo, S., "Space-Scanning Curves for Spatiotemporal Representation - Useful for Large Scale Neural Network Computing," *Proceedings of the International Joint Conference on Neural Networks*, Washington, D.C., January 1990, II-703.

19. Van den Bout, D. E., and Miller, T. K., "A Traveling Salesmen Objective Function That Works," *Proceedings of 1988 IEEE International Conference on Neural Networks*, II-299.

20. Wacholder, E., Han, J., and Mann, R. C., "An Extension of the Hopfield-Tank Model for Solution of the Multiple Traveling Salesmen Problem," *Proceedings of 1988 IEEE International Conference on Neural Networks*, II-305.

21. Xu, X., and Tsai, W. T., "An Adaptive Neural Algorithm for the Traveling Salesman Problem," *Proceedings of the International Joint Conference on Neural Networks*, Washington, D.C., January 1990, II-716.

22. Yen, W., and McLean, D., "Combining Heuristics for Optimizing a Neural Net Solution to the Traveling Salesman Problem," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 17-21, 1990.

APPENDIX A:
NEURAL NETWORK JOURNALS

Title: Neural Networks
Publisher: Pergamon Press
Address: Pergamon Journals Inc.
Fairview Park,
Elmsford, New York 10523

Official journal of the International Neural Network Society (INNS). Contains original contributions, invited review articles, letters to editors, invited book reviews, editorials, announcements, INNS news, and software surveys. The contributed papers are generally of high quality.

Title: Neural Computation
Publisher: MIT Press
Address: MIT Press Journals
55 Hayward Street
Cambridge, MA 02142-9949
Phone: (617) 253-2889

Combination of reviews, views, original contributions, and letters. Articles in this journal are usually short, but the quality is good.

Title: IEEE Transaction on Neural Networks
Publisher: Institute of Electrical and Electronics Engineers (IEEE)
Address: IEEE Service Center
445 Hoes Lane
P.O. Box 1331
Piscataway, NJ 08855-1331
Tel: (201) 981-0060

Devoted to the science and technology of neural networks which disclose significant technical knowledge, exploratory developments and applications of neural networks from biology to software and hardware. Emphasis is on artificial neural networks. Specific aspects include self organizing systems, neurobiological connections, network dynamics and architecture, speech recognition, electronic and photonic implementation, robotics, and controls. Includes letters concerning new research results.

Title: Journal of Neural Network Computing, Technology, Design, and Applications
Publisher: Auerback Publishers
Address: Auerback Publishers
210 South Street
Boston, MA 02111-9812
Tel: (800) 950-1216

About 5 indepth articles per issue; bookshelf section provides a several page introduction to a specific topic and a list of references for further reading on that topic; software reviews. The

targets of this journal are researchers and managerial personals working in industry and governmental agencies.

Title: International Journal of Neural Systems
Publisher: World Scientific Publishing
Address: World Scientific Publishing Co.
687 Hartwell Street
Teaneck, NJ 07666
Tel: (201) 837-8858

The International Journal of Neural Systems is a quarterly journal which covers information processing in natural and artificial neural systems. It publishes original contributions on all aspects of this broad subject that involves physics, biology, psychology, computer science, and engineering. Contributions include research papers, reviews and short communications. The journal presents a fresh undogmatic attitude towards this multidisciplinary field with the aim to be a forum for novel ideas and improved understanding of collective and cooperative phenomena with computational capabilities.

Title: Connection Science: Journal of Neural Computing, Artificial Intelligence and Cognitive Research
Publisher: Carfax Publishing
Address: Carfax Publishing Company
P. O. Box 25
Abingdon, Oxfordshire
OX14 3UE, UK

Title: Concepts in NeuroScience
Publisher: World Scientific Publishing
Address: World Scientific Publishing Co.
687 Hartwell Street
Teaneck, NJ 07666
Tel: (201) 837-8858

Title: Neurocomputers
Publisher: Gallifrey Publishing
Address: Gallifrey Publishing
PO Box 155
Vicksburg, Michigan 49097
Tel: (616) 649-3772

Title: Complex System
Publisher: Complex Systems Publications
Address: Complex Systems Publications, Inc.
P.O. Box 6149
Champaign, IL 61821-8149

Title: AI EXPERT
Publisher: Miller Freeman Publications

Address: 500 Howard St.
San Francisco, CA 94105
Tel: (415) - 397-1881

A popular AI magazine and self-claimed as the magazine of artificial Intelligence in Practice. Publishes introductory articles on neural networks such as the series: "Neural Networks Primer" and "Using Neural Nets" by Maureen Caudill.

Title: Neural Network Review
Publisher: Lawrence Erlbaum Associates (LEA)
Address: Lawrence Erlbaum Associates Inc.
365 Broadway
Hillsdale, NY 07642

Review Journal. Reviews of book, products, selected papers from other journals; announcements for news items, books, journals, and conference proceedings; copies of table of contents for several journals and proceedings. Accompanying each review article, it usually presents an original author's response.

Title: Neural Network News
Publisher: AIWeek Inc.
Address: Neural Network News
2555 Cumberland Parkway, Suite 29,
Atlanta, GA 30339
Tel: (404) 434-2187

A commercial newsletter. It presents reviews of neural network conferences, new products, and research activities in the United States, Europe, and Japan.

APPENDIX B:
PUBLICLY AVAILABLE SIMULATORS FOR ARTIFICIAL NEURAL NETWORKS

This compiled listing on publicly available software for neural networks simulation is based on information from Neural Network Digest - an electronic bulletin board on Internet. Information regarding the purpose and availability of each software is not guaranteed to be completely correct because the validity of each ftp address has not been verified.

BPS - George Mason University Back Prop Simulator
Current version is 1.01 (Nov., 1989)

A special-purpose simulator for backpropagation and a BP speedup technique called "gradient correlation." Available via anonymous ftp from gmuvax2.gmu.edu (129.174.1.8). Distributed as executable for VAX 8530 under Ultrix 3.0, and versions for 8088 based IBM PC, and 80286/386 IBM PC machines. Includes examples and a tutorial document. Source code license is available.

Contact:
Eugene Norris
Computer Science Department
George Mason University
Fairfax, Virginia 22032
Email: enorris@gmuvax2.gmu.edu
Tel: (703) 323-2713

MIRRORS/II — Maryland MIRRORS/II Connectionist Simulator
A general-purpose connectionist simulator. MIRRORS/II is implemented in Franz Lisp and will run under Opuses 38, 42, and 43 of Franz Lisp on UNIX systems. It is currently running on a MicroVAX, VAX and SUN 3.

To obtain this simulator you must sign an institutional site license. A license for individuals is not acceptable. The only costs incurred are for postage for a printed copy of the manual and tape cartridge (you send your own 1/4" cartridge or TK50 cartridge to them, if desired.) Instructions for obtaining the software via ftp are returned to you upon receipt of the license agreement. To obtain a copy of the license send your U. S. Mail address via e-mail to: mirrors@cs.umd.edu.

Or by U.S. Mail to:

Lynne D'Autrechy
University of Maryland
Department of Computer Science
College Park, MD 20742

NeurDS - The Neural Design and Simulation System.
Current Version is 3.1 (May, 1989)

A general purpose simulator. The system is licensed on a no-fee basis to educational institutions by Digital Equipment Corporation. To obtain information, send your U. S. or electronic mail address to:

Max McClanahan
Digital Equipment Corporation
1175 Chapel Hills Drive
Colorado Springs, Colorado 80920-3952
Email: mcclanahan%cookie.dec.com@decwrl.dec.com

You should receive instructions on how to obtain a copy of the manual and copies of the license agreement.

The NeurDS system will run on any Digital platform including Vax/VMS, Vax/Ultrix, and DECsystem/Ultrix. A graphics terminal is not required to support the window interface. Specific models are described using a superset of the C programming language, and compiled into a simulator form. This simulator can accept command scripts or interactive commands. Output can take the form of a window-type environment on VT100 terminals, or nonwindow output on any terminal.

FULL — Fully connected temporally recurrent neural networks.
A demonstration network described in "Learning State Space Trajectories in Recurrent Neural Networks."

The author (Barak Pearlmutter, bap@f.gp.cs.cmu.edu) describes this as "a bare bones simulator for temporally recurrent neural networks" and claims that it should vectorize and parallelize well. It is available for ftp from doghen.boltz.cs.cmu.edu. Login as "ftpguest", password "aklisp". Be sure to ftp as binary for the file "full/full.tar.Z" (you must either use a directory named full on your local machine, or use "get" and let it prompt you for remote and local file names). Do not attempt to change directories. It is copyrighted and is given out for academic purposes.

GRADSIM Connectionist Network Simulator.
A special-purpose simulator specifically designed for experiments with the temporal flow model. Latest Version 1.7.

In C, implementations on VAX (VMS & Ultrix), Sun, and CYBER are mentioned. Includes an excellent article with references. The simulator is available for anonymous ftp from ai.toronto.edu (128.100.1.65). For information contact:

Raymond Watrous
Department of Computer Science
University of Toronto
Toronto, Ontario M5S 1A4
Email: watrous@ai.toronto.edu

GENESIS - GEneral NEural Simulation System with
XODUS - X-windows Output and Display Utility for Simulations
A general simulator. Currently Beta-Test Version, 1990. From the release announcement (January 1990 by Jim Bower

Full source for the simulator is available via ftp from genesis.cns.caltech.edu (131.215.135.64). To acquire FTP access to this machine it is necessary to first register for distribution by using telnet or rlogin to log in under user "genesis" and then follow the instructions. When necessary, tapes can be provided for a handling fee of US$50. Those requiring tapes should send requests to genesis-req@caltech.bitnet. Any other questions about the system or its distribution should also be sent to this address. GENESIS and XODUS are written in C and run on SUN and DEC graphics work stations under UNIX (version 4.0 and up), and X-windows (version 11). The software requires 14 meg of disk space and the tar file is approximately 1 meg.

The current distribution includes full source for both GENESIS and XODUS as well as three tutorial simulations (squid axon, multicell, visual cortex). Documentation for these tutorials as well as three papers describing the structure of the simulator are also included. As described in more detail in the "readme" file at the ftp address, those interested in developing new GENESIS applications are encouraged to become registered members of the GENESIS users group (BABEL) for an additional one time $200 registration fee. As a registered user, one is provided documentation on the simulator itself, access to additional simulator components, bug report listings, and access to a user's bulletin board.

SunNet
A generalized simulator. Version 5.5.2.4 currently.

Available for anonymous ftp from boulder.colorado.edu (128.138.240.1). While this program was obviously written for Sun workstations (versions for Suntools and the X-window environment), the documents list other configurations. These include a nongraphic version which runs on "any UNIX machine," and versions which run on an Alliant or UNIX machine and send data to a graphics support program running on a Sun workstation. It is very easy to install. A mailing list exists for users of the simulator.

RCS - The Rochester Connectionist Simulator
A general simulator. Version 4.2 currently.

Available for anonymous ftp from cs.rochester.edu (192.5.53.209). Tapes may be purchased (1600 bpi 1/2" reel or QIC-24 Sun 1/4" cartridge) from:

Peg Meeker
Computer Science Department
University of Rochester
Rochester, New York 14627

C source code is provided, including a graphic interface which may function under X Windows or SunView on Sun Workstations. A wide variety of Unix machines are supported, and the simulator may be used without the graphics interface. A version for the MacIntosh is included in the distribution. Mailing lists exist for users and bug reports.

SFINX — Structure and Function in Neural Connections
A General Simulator. Version 2.0 (November 1989)

In order to ftp this simulator, a license agreement must be submitted. Upon receipt of this agreement, instructions and the password to ftp the software are made available. To obtain the license write:

Machine Perception Laboratory
Computer Science Department
University of California
Los Angeles, CA 90024

This system requires color to operate the graphics interface, but may be operated without graphics. Support for Sun, Ardent Titan, HP 300, and IBM PC RT machines is specifically mentioned, but other Unix platforms should function as well. Specific graphics support is provided for Matrox VIP 1024, Imagraph AGC-1010P, HP Starbase and X Windows.

Mactivation
A specialized simulator for investigating associative memory using the delta rule and Hebbian Learning. Version 3.3 currently.

A public domain version is available for anonymous ftp from the University of Colorado at Boulder (boulder.colorado.edu, 128.138.240.1) or possibly by contacting the author.

Mike Kranzdorf
University of Colorado
Optoelectronic Computing Systems Center
Campus Box 525
Boulder, Colorado 80309-0525
Email: mikek@boulder.colorado.edu

Future versions will probably not be public domain, but will be available from Oblio, Inc., 5942 Sugarloaf Road, Boulder, Colorado 80309. Provided as executable for the Apple MacIntosh.

PDP Simulators
Several special purpose simulators are provided with the following book:

McClelland, J. L., and David.E. Rumelhart, *Explorations in Parallel Distributed Processing,* Vol. III, Cambrige: MIT Press, 1988.

The simulators were written in C, and versions for both the IBM PC and the MacIntosh exist.

Hopfield-style Network Simulator
A Special Purpose simulator for experimentation with the Hopfield-style network.

Software is available by e-mail upon request from the author, Arun Jagota. It is written in C and should be useful on 32-bit Unix machines, and a MSDOS version is also supplied. Arun's email address is jagota@cs.buffalo.edu.

APPENDIX C:
NEURAL NETWORK BOOKS AND PROCEEDINGS

This section lists some of the well known publications on neural networks, connectionist systems, computational psychology, genetic algorithms, vision and perception, and proceedings of conferences on neural networks, to give a historical perspective on the development of neural network research, to provide fundamental materials for beginners embarking on this field, and to provide researchers with the state-of-the-art publications from recent conferences dedicated to the research and application of neural networks.

1. *Proceedings of the IEEE First International Conference on Neural Networks*, IEEE, New York, June 1987.
2. *Proceedings of the IEEE International Conference on Neural Networks*, IEEE, New York, June 1988.
3. *Proceedings of the 1988 Connectionist Models Summer School*, D. Touretzsky, G. Hinton, and T. Sejnowski, Eds., Carnegie Mellon University, Morgan Kaufmann Publishers, San Mateo, CA, 1989.
4. *Proceedings of the International Joint Conference on Neural Networks*, Co-sponsored by IEEE and the International Neural Network Society, Washington, D. C., 1989.
5. *Proceedings of the International Joint Conference on Neural Networks*, Co-sponsored by IEEE and the International Neural Network Society, Washington, D. C., 1990.
6. *Proceedings of the International Joint Conference on Neural Networks*, Co-sponsored by IEEE and the International Neural Network Society, San Diego, 1990.
7. *Proceedings of the First International Conference on Genetic Algorithms*, J. J. Grefenstettee (Ed.), Lawrence Erlbaum Publishers, Hillsdale, NJ, 1987.
8. *Proceedings of the Second International Conference on Genetic Algorithms*, J. J. Grefenstettee, (Ed.), Lawrence Erlbaum Publishers, Hillsdale, NJ, 1988.
9. *Proceedings of the Third International Conference on Genetic Algorithms*, J. D. Schaffer (Ed.), Morgan Kaufmann Publishers, San Mateo, CA, 1990.
10. Ackley, D., *A Connectionist Machine for Genetic Hillclimbing*, Kluwer Academic Publishers, 1987.
11. Aleksander, I. (Ed.), *Neural Computing Architecture*, The MIT Press, Cambridge, MA, 1989.
12. Amari, S. I., and Arbib, M. (Eds.), *Competition and Cooperation in Neural Networks*, Springer-Verlag, New York, 1982.
13. Anderson, J. and Lehmkuhle, S. (Eds.), *Synaptic Modification, Neuron Selectivity, and Nervous System Organization*, Lawrence Erlbaum, Hillsdale, NJ, 1985.
14. Anderson, D. Z. (Ed.), *Neural Information Processing Systems*, American Institute of Physics, 1988.
15. Arbib, M. A., Caplan, D., and Marshall, J. C. (Eds.), *Neural Models of Language processing*, The Academic Press, New York, 1982.

16. Arbib, M. A., *Brains, Machines and Mathematics*, 2nd edition, Springer-Verlag, New York, 1987.

17. Basar, E., Flohr, H., Haken, H., and Mandel, A. J. (Eds.), *Synergetics of the Brain*, Springer-Verlag, New York, 1983.

18. Beck, J., Hope, B., and Rosenfeld, A. (Eds.), *Human and Machine Vision*, Academic Press, New York, 1983.

19. Caianiello, E. R. (Ed.), *Parallel Architectures and Neural Networks*, World Scientific, Singapore, 1989.

20. Carbonell, J. G. (Ed.), *Machine Learning: Paradigms and Methods*, The MIT Press, Cambridge, Massachusetts, 1990.

21. Casti, J. L., *Alternate Realities: Mathematical Models of Nature and Man*, Wiley Interscience, 1989.

22. Caudill, M., and Butler, C., *Naturally Intelligent Systems*, The MIT Press, Cambridge, Massachusets, 1990.

23. Commons, M. L., Grossberg, S., and Staddon, J. E. R. (Eds.), *Neural Network Models of Conditioning and Action*, Lawrence Erlbaum, Hillsdale, NJ, 1991.

24. Cornsweet, T. N., *Visual Perception*, Academic Press, New York, 1970.

25. Davis, L. (Ed.), *Genetic Algorithms and Simulated Annealing*, Morgan Kaufmann Publishers, Los altos, CA, 1987.

26. Davis, J., Newburgh, R., and Wegman, E. (Eds.), *Brain Structure, Leaning, and Memory*, AAAS Symposium Series, 1987.

27. Dayhoff, J. E., *Neural Network Architectures: An Introduction*, Van Nostrand Reinhold, New York, 1990.

28. Denker, J. S. (Ed.), *Neural Networks for Computing*, American Institute of Physics, 1988.

29. Durbin, R., Miall, C., and Mitchson, G., *The Computing Neuron*, The Academic Press, New York, 1989.

30. Eberhart, R. C., and Dobbins, R. W. (Eds.), *Neural Network PC Tools*, The Academic Press, New York, 1990.

31. Eckmiller, R., Hartmann, G., and Hauske, G. (Eds.), *Parallel Precessing in Neural Systems and Computers*, North-Holland, Amsterdam, The Netherlands, 1990.

32. Freeman, J., and Skapura, D., *Artificial Neural Systems; Theory and Practices*, The Academic Press, New York, 1991.

33. Freeman, W. J., *Mass Action in the Nervous System*, The Academic Press, New York, 1975.

34. Garner, W. R., *The Processing of Information and Structure*, Lawrence Erlbaum associates, Inc., Hillsdale, NJ, 1974.

35. Goldberg, D. E., *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley Publishers, New York, 1989.

36. Grossberg, S. (Ed.), *Mathematical Psychology and Psychophysiology*, American Mathematical society, Providence, RI, 1981.

37. Grossberg, S., *Studies of Mind and Brain: Neural Principles of Learning, Perception, Development, Cognition and Motor Control*, Reidel Press, Boston, 1982.

38. Grossberg, S., and Kuperstein, *Neural Dynamics of Adaptive Sensory Motor Control: Ballistic Eye Movements*, North-Holland, Amsterdam, 1986.

39. Grossberg, S., *The Adaptive Brain, I: Cognition, Learning, Reinforcement and Rhythm*, North-Holland, Amsterdam, 1987.

40. Grossberg, S., *The Adaptive Brain, II: Vision, Speech, Language, and Motor Control*, North-Holland, Amsterdam, 1987.

41. Grossberg, S. (Ed.). *Neural Networks and Natural Intelligence*, The MIT Press, Cambridge, Massachusets, 1988.

42. Hawkins, R. D., and Bower, G. H. (Ed.), *Computational Models of Learning in Simple Neural Systems*, The Academic Press, New York, 1989.

43. Hebb, D. O., *The Organization of the Behavior*, John Wiley and Sons, New York, 1949.

44. Hertz, J., Krogh, A., and Palmer, R., *Introduction to the Theory of Neural Computation*, The Academic Press, New York, 1990.

45. Hinton, G. E., and Anderson, J. A. (Eds.), *Parallel Models of Associative Memory*, Lawrence Erlbaum, Hillsdale, NJ, 1981.

46. Janko, W. H., Roubens, M., and Zimmermann, H.-J. (Eds.), *Progress in Fuzzy Sets and Systems*, Theory And Decision Library, Kluwer Academic Publishers, The Netherlands, 1990.

47. Khanna, T., *Foundations of Neural Networks*, Addison-Wesley, New York, 1990.

48. Koch, C. (Ed.), *Computation and Neural Systems*, The Academic Press, New York, 1990.

49. Kohonen, T., *Associative Memory: A System Theoretical Approach*, Springer-Verlag, New York, 1977.

50. Kohonen, T., *Self-Organization and Associative Memory*, Springer-Verlag, New York, 1984.

51. Levine, D. S., *Introduction to Neural and Cognitive Modeling*, Lawrence Erlbaum, Hillsdale, NJ, 1990.

52. Levine, D. S., and Leven, S. J. (Eds.), *Motivation, Emotion, and Goal Direction in Neural Networks*, Lawrence Erlbaum, Hillsdale, NJ, 1990.

53. MacGregor, R. J., *Neural and Brain Modeling*, The Academic Press, New York, 1987.

54. Marr, D., *Vision*, San Francisco, W. H. Freeman, 1982.

55. Mead, C., *Analog VLSI and Neural Systems*, The Academic Press, New York, 1989.

56. Mel, B., *Connectionist Robot Motion Planning*, The Academic Press, New York, 1990.

57. Minsky, M., and Papert, S., *Peceptrons: An Introduction To Computational Geometry*, The MIT Press, Cambridge, Massachusetts, 1969, and 1988.

58. Pao, Y. -H., *Adaptive Pattern Recognition and Neural Networks*, Addison-Wesley, New York, 1989.

59. Parten, C., harston, C., Maren, A., and Pap, R., *Handbook of Neural Computing Applications*, Academic Press, 1990.

60. Rosenblatt, F., *Principles of Neurodynamics*, Spartan Books, Washington, D. C., 1962.

61. Rumelhart, D. E., McClelland, J. L., and the PDP Research group, *Parallel Distributed Processing: Exploration in the Microstructure of Cognition*, Vol. 1-3, The MIT Press, 1986.

62. Simpson, P. K., *Artificial Neural Systems: Foundations, Paradigms, Applications and Implementations*, Pergamon Press, New York, 1990.

63. Soucek, B., and Soucek, M., *Neural and Massively Parallel Computers*, Wiley, 1989.

64. Taylor, J. G., and Mannion, C. L. T. (Eds.), *New Developments in Neural Computing*, Adam Hilger, New York, 1989.

65. Touretzky, D. (Ed.), *Advances In Neural Information Processing Systems 1*, Morgan Kaufmann, CA, 1989.

66. Touretzky, D. (Ed.), *Advances in Neural Information Processing Systems 2*, Morgan Kaufmann, CA, 1990.

67. Vemuri, V. (Ed.), *Artificial Neural Networks: Theoretical Concepts*, Computer Society Press, IEEE, New York, 1988.

68. Von Newmann, J., *The Computer and the Brain*, New Haven, Yale University Press, 1958.

69. Wasserman, P., *Neural Computing: Theory And Practice*, Van Nostrand Reinhold, New York, 1990.

70. Wasserman, P. D., and Oetzel, R. M., *Neuralsource: The Bibliographic Guide To Artificial Neural Networks*, Van Nostrand Reinhold, New York, 1990.

71. Yovits, M. C., Jacobi, G. T., and Goldstein, G. D. (Eds.), *Self-Organization Systems*, Spartan Books, Washington, D. C., 1962.

72. Zornetzer, S. F., Davis, J. L., and Lau, C. (Eds.), *An Introduction to Neural and Electronic Networks*, Academic Press, New York, 1990.

DISTRIBUTION

Chief of Engineers
ATTN: CEHEC-IM-LH (2)
ATTN: CEHEC-IM-LP (2)
ATTN: CERD-L

Fort Belvoir, VA 22060
ATTN: CECC-R
ATTN: Engineer Topographic Lab

CECRL, ATTN: Library 03755

CEWES, ATTN: Library 39180

Defense Technical Info. Center 22304
ATTN: DTIC-FAB (2)